DOCUMENT RESUME

ED 312 894 FL 018 222

AUTHOR Meadow, Anthony, Ed.
TITLE Newsletter for Asian and Middle Eastern Languages on Computer, Volume 1, Numbers 1 and 2.
PUB DATE 85
NOTE 70p.
PUB TYPE Collected Works - Serials (022)
JOURNAL CIT Newsletter for Asian and Middle Eastern Languages on Computer; v1 n1-2 Jan-Sep 1985

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS Alphabets; *Computers; *Computer Software; Diacritical Marking; Japanese; Uncommonly Taught Languages; *Word Processing
IDENTIFIERS Middle East; South Asian Languages; Transliteration

ABSTRACT

Numbers 1 and 2 of the first volume of the newsletter contains an editor's page and the following articles: "Diacritics on Wordstar: South Asian Language Transliteration without Customized Software" by Tony Stewart; "The Universal Typewriter" by David K. Wyatt and Douglas S. Wyatt; "Multi-Lingual Word-Processing Systems: Desirable Features from a Linguist's Point of View" by Lloyd Anderson; "A Note on the Production of Macrons in Transliterated Japanese" by Jay Rubin; and "Indian Fonts on the Macintosh" by George Hart. Other regular newsletter features include organization news; reviews of books, journals, articles, and products; a calendar of events; and announcements and inquiries. (MSE)

**********
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
**********

ED312894

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

# Newsletter for Asian and Middle Eastern Languages on Computer

Featuring articles and news about Asian and Middle Eastern languages on computer

ISSN 0749-9981

Volume 1, Number 1 January, 1985

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
meadow, A.
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Editor's Page:

Introduction to the Newsletter
Anthony Meadow ..... 1

Articles:

Diacritics on Wordstar: South Asian Language Transliteration without Customized Software
Tony Stewart ..... 4

The Universal Typewriter
David K. Wyatt and Douglas S. Wyatt ..... 14

Multi-Lingual Word-Processing Systems: Desirable Features from a Linguist's Point of View
Lloyd Anderson ..... 28

Regular Departments:

Reviews:
Organizations ..... 31
Books ..... 33
Journals ..... 34
Articles ..... 36
Calendar of Events ..... 38
Announcements and Queries ..... 39

FL018222



BEST COPY AVAILABLE

## Policies

The **Newsletter for Asian and Middle Eastern Languages on Computer** is published by Bear River Systems. The entire contents of the Newsletter are copyright 1985 by Bear River Systems. Contents of the Newsletter may be reproduced for non-profit use provided credit is given to both the author and Bear River Systems. The credit line should indicate that the material was first published in the Newsletter for Asian and Middle Eastern Languages on Computer, PO Box 1021, Berkeley, CA 94701. For any other use of material, please contact the editor.

A one year subscription costs $10 throughout the world. American subscriptions help subsidize subscriptions for the rest of the world. A year's subscription includes the entire volume which consists of four issues per year. Please send a check or money order in US funds made payable to "Bear River Systems" to the following address:

Newsletter for Asian and Middle Eastern Languages
on Computer
Bear River Systems
PO Box 1021
Berkeley, CA 94701
USA

The newsletter is produced on a part-time basis and there is not sufficient time to deal with various forms of red-tape. Please send a check or money order, rather than a purchase order.

The newsletter is not refereed.

Exchanges with other newsletters will be considered. Please contact the editor.

## Submitting Articles

The newsletter is produced by reproduction and cannot be easily typeset because of the subject matter. Therefore, when submitting articles, please send them in "camera-ready" copy form. When printing your article out, please therefore use a new ribbon to ensure better print quality.

Please send two copies of the article to:

Editor, Newsletter for Asian and Middle Eastern Languages
on Computer
Center for South and Southeast Asian Studies
260 Stephens Hall
University of California
Berkeley, CA 94720
USA

# Introduction to the Newsletter

Welcome to the Newsletter for Asian and Middle Eastern Languages on Computer This is the first of many issues of the only publication dealing with Asian and Middle Eastern languages on computer. I started this newsletter to help bring together all people who are working with any of the Asian or Middle Eastern languages on computer because there is so little available on this topic from any other journal or organization. This newsletter is published with some help from the Center for South and Southeast Asian Studies at the University of California at Berkeley, although there is no formal connection between the Center and this newsletter. Sole responsibility for the contents of the newsletter rests with the editor.

We hope to cover a large geographical area, including all of Asia and the Middle East. There are certainly enormous differences between the writing systems of Japan, India and Saudi Arabia. Although these languages use writing systems that differ from each other, each system is certainly different from English. Much can be learned from the techniques used to put, for example, Japanese on an Apple II, even for someone working with Sanskrit. We are still in the early stages of making this happen and it is still an innovative programmer who can put Chinese or Korean or Tamil on an IBM PC. In the short history of computers, almost all attention has been placed on working with English and other European languages Only recently has it been both possible and affordable to use non-western writing systems on computers. None the less, more and more Asian and Middle Eastern scripts are making it into computers, and this newsletter is here to help anyone interested to keep up with these exciting events.

All the computers mentioned so far are microcomputers. The reason for this is that we are now in an age when computing power and access are available in a box that will comfortably fit onto a desk top for a small amount of money. The newer microcomputers in particular have enough graphics capabilities that non-Western writing systems can be easily displayed This is not to say that minicomputers or mainframe computers can't be used, it's just that more work in this area seems to be occurring on micros. Nevertheless, we will cover the field no matter what the computer involved.

The potential audience for this newsletter is unusually broad, both in knowledge of computers and in possible applications of the programs. Some of the subscribers include scholars of the various languages, librarians, historians, linguists, business people and diplomats. Some are professional computer scientists and others have only recently begun using computers. This makes for a wide variety of articles, as can be seen in the articles in this first issue.

This newsletter can only exist on the contributions of its readers It seems that in the last couple of years, many people have ventured into using computers for word processing and other such tasks. It is still no easy task to use Asian and Middle Eastern languages on computer, so if you've found a better way, or you have suggestions for programmers on how it should be approached, please submit an article on it.

One of the topics that we'll be dealing with in future issues is computer networks. This will become a more and more important method of communications, for individuals as well as organizations. I am most interested in receiving

manuscripts electronically. If you are interested in submitting your article through CompuServe or through some other network, please contact the editor.

This newsletter is being produced for a minimal profit. My time spent editing the newsletter is not compensated for, at least financially. The subscription money will be used to pay for paper, printing, envelopes, mailing labels and postage. I don't expect to make any profit, but I also don't want to lose money on this venture. I have reluctantly increased the subscription fee to $10 per year. This is for several reasons. The size of the newsletter is about twice as large as I had originally expected. Postage rates are scheduled to go up shortly after this issue will be mailed. The newsletter will be printed rather than photocopied. If you have already paid for your subscription, then you got a bargain! Everyone else must pay the new rate of $10 per year. Note that this rate is for the entire world. The American standard of living is higher than the rest of the world, with perhaps a few countries being exceptions. The $10 that an American pays generates a small profit. This profit will be used to subsidize the cost of airmail postage to the rest of the world.

The majority of this newsletter was written and edited on an Apple Lisa 2 computer using the Lisa 7/7 Office System. It was printed on an Apple DMP II printer.

A large number of people have helped in various ways to get this newsletter off the ground. I would like to thank the following people, hoping that I haven't left anyone off: first from the University of California at Berkeley: Bruce Pray and Barbara Guerlin of the Center for South and Southeast Asian Studies, Ken Logan of the South and Southeast Asian Library, Sally Sutherland and George Hart of the Department of South and Southeast Asian Library, James Agenbroad of the US Library of Congress and James Nye, now at the University of Chicago. Lastly, this newsletter would never have gotten out if it were not for the assistance of my wife, Diana.

It has taken longer than I ever thought to produce the first issue, but now that it is out, future issues will take less time to produce. Thank you all for being so patient.

## In this Issue

There are three articles in this issue. The first, on using Wordstar to produce diacritics, is by Tony Stewart. This article is a revision of an early draft that he was circulating in Chicago. Tony is a doctoral student in the Department of South Asian Languages and Civilizations at the University of Chicago. He is interested in the history of South Asian Religion. He can be reached at: 713 W. Barry Avenue, Apt 2-N, Chicago, IL 60657.

The second article, entitled "The Universal Typewriter", is by Professor David Wyatt and one of his sons. This article gives a mildly non-technical overview of some software that the two developed for working with Southeast Asian languages and English. While the software now only runs on a particular brand of CP/M-80 computer, it is being moved to the IBM PC. David Wyatt is Professor of History at Cornell University. His interests are in the history of Southeast Asia, especially Thailand and Laos. He can be reached at: Department of History, Cornell University, McGraw Hall, Ithaca, NY 14853.

The last article, on multi-lingual word-processors, was written by Lloyd Anderson. This article is a slightly modified version of a paper he has been circulating among linguists. Lloyd is a typological linguist and has recently become involved with the national standards committee for character sets. He can be reached at: Ecological Linguistics, 316 A Street, S.E., Washington, D.C. 20003.

Regular features are a calendar of events and a section for announcements and queries.

The reviews section in this issue covers organizations, journals, books and articles, but no hardware or software. The next issue will have a large set of listings of software and hardware. All reviews were written by the editor.

Anthony Meadow

## DIACRITICS ON WORDSTAR®
## SOUTH ASIAN LANGUAGE TRANSLITERATION WITHOUT CUSTOMIZED SOFTWARE

Tony K. Stewart
University of Chicago

Diacritics for transliterating the many South Asian languages **can** be produced on your microcomputer with off-the-shelf word processing software. Of the many packages, **Wordstar®** handles the diacritics "problem" with perhaps the greatest facility, although not necessarily the greatest ease, of any of the more readily available word processing programs. Among the products which were personally tested, **Benchmark®**, **Word III®**, **The Final Word®**, and **WPS-80™** word processing programs proved themselves capable of producing some manner of diacritics, but with rather severe limitations in the production process or final copy. The most common drawback was the placement of the diacritic mark itself: the diacritic mark tended to fall in full half-line increments above or below the character in question.[1] This resulting copy proved less desirable in most cases than manually typed copy using an IBM **Selectric™** typewriter with European ball, or some equivalent. **Wordstar**, however, possesses what appears to be a unique feature among off-the-shelf word processing programs: the ability to raise or lower the superscript and subscript to the desired height. Of course your printer must be capable of such fine adjustments (more on that later), but the program itself will handle virtually any requirement for diacritics in the transliteration of Sanskritic or Dravidian languages. The same process will apply for creating diacritics to represent other Asian languages, the only limitation being the symbols required on the printwheel or fashioned on a dot matrix. To describe the procedure for using **Wordstar** in the production of diacritics, we shall consider four broad topics: 1. How to create the diacritics; 2. printer requirements; 3. additional features to facilitate editing and printing with diacritics; 4. user definable function keys.

---

[1] It should be noted that I did not have the opportunity to test **Wordperfect®** by Satellite Software International, which superscripts and subscripts one third of a line, a distance that should allow for acceptable placement of the diacritics.

* * *

**Wordstar®** is a registered trademark of MicroPro® International; **Benchmark®** is a registered trademark of Metasoft Corporation; **Word III®** is a registered trademark of Samna; **The Final Word®** is a registered trademark of Mark of the Unicorn; **WPS-80™** is a registered trademark of Exceptional Business Solutions; **Selectric™** is a registered trademark of IBM.

## I. HOW TO CREATE TRANSLITERATION DIACRITICS ON WORDSTAR

### A. COMMANDS:

All diacritics can be created using only three commands: superscript, subscript, and overstrike. All three commands are located in the "print" menu of **Wordstar:**

1. ^P^T = superscript
2. ^P^V = subscript
3. ^P^H = overstrike (or backspace)

The commands are used either singly or in combination, depending on the diacritic mark desired. In the case of the superscript (^P^T) and the subscript (^P^V), you must always remember to turn the command off after the mark has been typed, for these two commands are "toggles." A toggle is either "on" or "off" and will remain in that state until reversed. The overstrike command (^P^H) is not a toggle and will only affect the subsequent character.

### B. SEQUENCE:

In standard transliteration schemas for most Indian languages, the macron, denoting a long vowel requires the superscript (^P^T) and overstrike (^P^H). The dot or period below a character, generally denoting a retroflex consonant or 'vocalic r' requires the subscript (^P^V) and overstrike (^P^H) in combination. The tilde for nasals requires only the overstrike (^P^H) because it is already located in the correct position to print above a character and needs only to be backspaced. The accent, for the sibilant is the same as the tilde in that it only requires the use of an overstrike or backspace command (^P^H).

Remembering that the superscript and subscript commands must be turned on prior to the diacritic mark and turned off immediately thereafter, the sequence of actual keystrokes for producing examples of the four marks looks like this:

1. macron = ^P^T-^P^T^P^Ha producing /ā/
2. dot or period = ^P^V.^P^V^Hd producing /ḍ/
3. tilda = n^P^H~ producing /ñ/
4. accent = s^P^H´ producing /ś/

Note that those operations requiring superscript or subscript follow a sequence which types the diacritic mark first, then the character itself, while the tilde and accent reverse that procedure.[1]

C. ONSCREEN:

You should note at this point, however, that what you see on your screen will be different from the above in several respects. First, when you invoke the print menu, the ^P command does not stay on the screen, but the following command (either ^T,^V, or ^H) will show on the screen. Second, all of the many versions of **Wordstar** tested on different computers displayed the diacritic in a separate column beside the character, rather than on top of or below it. Consequently, you should expect a foreign word such as Rādhā to appear on the screen as

R^T-^T^Hadh^T-^T^Ha

before removing the commands.

The print commands can be removed from the screen (but remain in place) by using the "**O**nscreen Print **D**isplay" command (^O^D). After suppressing the print commands with the ^O^D, your screen should look like this: R-adh-a. A liberal use of the ^O^D command will facilitate your editing and proofreading as you type. Being a toggle, the ^O^D command can be used as often as needed to check the spelling of the word, for in practice, amidst the clutter of print commands, the actual letters and diacritics of the word can get confused or omitted.

D. SUPERSCRIPT/SUBSCRIPT ROLL (.SR N):

It is the ability to raise or lower the super/subscript in minute increments which sets **Wordstar** apart from other word-processing packages and makes it well suited for producing diacritics. To raise or lower the super/subscript use the dot command (.SR n). **Wordstar** can adjust this roll in 1/48" increments. The default value for the **S**uper/subscript **R**oll is 3/48" which is good for most applications (.SR 3). If

---

[1]Dividing the print commands into two separate steps--associating only one command with each printed part of the new character--eases the load on the printer. The diacritic mark itself is either superscripted or subscripted, with the accompanying character merely being backspaced and overstruck, as opposed to printing the character in a normal position and attempting simultaneously to backspace and super/subscript the diacritic. Few printers can tolerate the convolutions of the latter.

you are typing in all capitals, however, it may be useful to change to 4/48" (.SR 4) which is a full half-line. If you want to bring the marks closer to the character (which will work with only selected print wheels) 2/48" (.SR 2) is the limit.

The super/subscript roll (.SR n) is adjustable only on a per line basis, which simply means you cannot change it mid-line. Since most changes would be made for capitalized titles, etc., this generally presents no problem. Examples of the super/subscript roll dot comand show the flexibility:

(.SR 4) RĀDHĀ; KṚṢṆA

Rādhā; Kṛṣṇa

(.SR 3) RĀDHĀ; KṚṢṆA

Rādhā; Kṛṣṇa

(.SR 2) RĀDHĀ; KṚṢṆA

Rādhā; Kṛṣṇa

Purists will dislike the compromise, however, between the position of the macron and the position of the subscripted dot, since the proper placement of the former pushes the latter a bit too low under the character, while vice-versa crowds the macron onto the top of the character. This inability to switch super/subscript heights mid-line is the greatest weakness in the process.

**E. UNDERLINING AND BOLDFACING:**

Unlike a typewriter, which places the underscore directly below the word on the printed page, computers generally do not underline with a key-stroke, but with print commands. In **Wordstar** these commands have to be watched very carefully because the underscore toggle (^P^S) has to be switched off for the actual superscript or subscript, otherwise you will get a line under both of those as well. If you simply insert the underscore print command (^P^S) before the word and turn it off after the word, the result looks like this:

Rādhā; Kṛṣṇa

So, you must remember to turn underline command off before each super/subscripted diacritic and turn it on again before the following letter. The key strokes become quite involved and often lead to mistakes. For instance, the word Rādhā when properly underscored will involve the

following strokes (the boldface indicates the actual characters to be printed):

^P^S**R**^P^S^P^T-^P^T^P^H^P^S**adh**^P^S^P^T-^P^T^P^H^P^S**a**^P^S;

while the screen would look like this:

^S**R**^S^T-^T^H^S**adh**^S^T-^T^H^S**a**^S;

and it would print like this:

Rādhā.

Clearly, the effort involved is tedious and time consuming, but there is an alternative, which is just as effective and eliminates most of the hastle: **boldfacing.** Boldfacing effectively sets apart the word so that it is easily recognized as foreign and can be implemented by simply turning on the boldface print command (^P^B) at the beginning of the word or string of words and turning it off again at the desired end. Examples in boldface, rather than underline, look like this:

**Rādhā; Kṛṣṇa; jñāna; Kāśī.**

**F. CAUTIONS:**

There are three weaknesses in the **Wordstar** program which can lead to problems if not watched carefully. They are not insurmountable, rather simply restrictive.

1. When reforming paragraphs (^B), the macron (superscripted dash) can be read by the machine as a hyphen. This results in the breaking of a word at the macron with the following vowel appearing on the next line. This occurs because the superscript is only a print command, and does not differentiate the macron from a hyphen for any of the word-processing functions. This potential problem can be avoided by manually hyphenating problem words. Manual hyphenation requires close attention on the part of the typist, which is to say that the automatic repeating paragraph reform (^Q^Q^B) should probably be avoided. It is much easier to double-check one paragraph at a time as you move through a document, rather than reforming the entire document and going back to check word breaks.

2. Another major caution concerns the use of microjustification. Due to limitations on the power of **Wordstar**, the diacritic mark can be unevenly placed around the letter when microjustification is on. On words with a single diacritic this problem generally does not appear.

Similarly, on words with the diacritics sufficiently spread throughout, the problem frequently does not show. But on words with diacritics in close succession, the microjustification will on occasion jam up the sequence. For example, words with a succession of diacritics can look like this with microjustification turned on:

Kṛṣṇa or Caṇḍīdāsa

and like this when turned off:

Kṛṣṇa or Caṇḍīdāsa.

Your particular system may respond somewhat differently due to printer variation. However, the whole problem can be avoided by switching off microjustification either using a dot command (off = .UJ 0; on = .UJ 1), or by switching the microjustication off when you install the program. Note that microjustification is not the same as justifying the margins.

> If you do want justified right margins with microjustification turned off, the program will simply divide the number of spaces as evenly as possible between the words on the line (like this example), rather than by microdivisions, which appear much more even to the eye. Although this may not appear as even to the eye, it obviates the problem of diacritic placement and still allows for justification. For example, the words Rādhā and Vaiṣṇava are printed with proper placement of diacaritics.

The remaining option is, of course, simply not to justify the right margin (i.e. "ragged right margin"), keeping the microjustication permanently turned off, like this example.

As a final option to circumvent the problems mentioned above, you can trick **Wordstar** into doing the diacritics while justifying the margins and using microjustification (not to be confused with true proportional spacing, which **Wordstar** does not do). The latest versions of **Wordstar** (3.0-3.3) seem to have improved the quality of the microjustification, even though you can still have an occasional surprise. Each machine, however, will respond differently to **Wordstar** so you will have to experiment. I have found that by using a dot command for character width (.CW 11), I can right justify with microjustification turned on and still get the diacritics to come out properly underneath and overtop of the characters. This dot command pushes the characters together just a little, so that using a ten pitch wheel (i.e. Pica, normally .CW 12) you can get text to look almost like true proportional spacing. For a twelve pitch wheel (i.e. Elite, normally .CW 10), you can use .CW 9 for the same effect, but I do not care for that as much because it makes the

lines too long and therefore harder to read. The body of this text has been printed using a ten pitch Courier wheel with the character width set at .CW 11.

When you right justify and leave on the microjustification, the diacritics in words such as Kṛṣṇa or Vaiṣṇavas, and in less taxing words like Rādhā or brāhmaṇa turn out very nicely. It should be noted that by simply switching to the narrower pitch (.CW 11 for pica), the right margins are not as ragged when you do not right justify, nor does the eye pick up the discrepancy as quickly when you turn off the microjustification (.UJ Ø) and continue to right justify. In short, you will have to experiment with your own computer/printer combination to see what marriage of pitches, justification settings, etc. proves the most fruitful.

3. Single spacing: a possible problem. You may find that when you attempt to install your diacritics in single-spaced text, they may tend to bleed or run together vertically. The default values for **Wordstar** are set to assume six lines per inch (=66 lines/page), which is a line height of 8/48" (.LH 8). If you are experiencing this difficulty, you should adjust your line height to .LH 9 (=5.3 lines/inch) or .LH 1Ø (=4.8 lines/inch). When you change the line height, remember that you will have to change the page length (.PL n) to correspond. With .LH 9 or .LH 1Ø you may want to decrease the number of lines in the top margin (.MT n) and bottom margin (.MB n). Consult your **Wordstar** manual for the reasons and effects of this alterations, but as usual there is no substitute for experimentation. This text has been composed using a line height of .LH 9 and a page length of .PL 58.

## II. PRINTER REQUIREMENTS

### A. GENERAL:

Without an appropriate printer, the above commands are useless. In order to take full advantage of the power of **Wordstar** in creating diacritics, you must be extremely selective in choosing a printer. There are several minimal requirements:

1. The printer, whether daisy wheel or dot matrix, must be capable of printing the superscript/subscript and the overstrike or backspace. Good quality ink-jet and all laser printers are sufficiently sophisticated to handle any **Wordstar** command.

2. The printer must be capable of small line increments, preferably of 1/48". The fine increment is needed to accomodate the "superscript/subscript roll" (.SR n) explained above. Lower quality printers

will print the superscript/subscript only at a fixed height, most often 4/48" or one half line. If this is the case, then the diacritics will be too high or low, which not only defeats the advantages of using **Wordstar**, but completely eliminates the possibility for single spacing. If the printer is so limited, then many of the other word-processing programs mentioned in the opening paragraph that handle superscript/subscript would be equally acceptable.

**B. PRINTWHEELS:**

1. The printwheel for daisywheel printers must be carefully chosen. All printwheels will have the dash/hyphen and the period to create the macron and subscripted dot but not every printwheel will contain the tilde or acute accent. While the tilde does appear on many standard 96 character American printwheels, the acute accent will generally require a European printwheel, most of which possess a slanted apostrophe or acute accent in place of the vertical American apostrophe. (Ironically, almost all 96 character American printwheels have the grave accent [eg. /à/], which is almost never used in English, while the acute accent is often used in words of French derivation.) European printwheels can be obtained from your dealer.

2. Please note that some printwheels have both accents and the vertical apostrophe. Whether this is preferable is strictly a matter of taste, but on such printwheels the acute accent will be placed in a different position on the keyboard (using for example the Comm. System). Unless your particular computer has the various European alphabets built into ROM and you have programmed the change, your screen will not show the accent, but rather, will show whatever key is in the same place on the standard American keyboard. This can be disconcerting, but more importantly, it can lead to confusion and a lot of work if you change printwheels later.

3. Although not recommended, printwheels can be customized to specification. This is fairly costly. With the limited life of most printwheels, especially plastic (and most companies will only customize plastic), an untimely break can be devastating. If one chooses to go this route, it would probably be more effective to have the desired diacritic made up special and not fuss with the superscript/ subscript, etc. But one should be prepared to invest considerable money and time for testing before going ahead with this commitment.

4. If you cannot find a type font which is suitable among the European printwheels, you can use a standard American printwheel (with the vertical apostrophe in place of the slanted apostrophe or accute accent) and simply insert the "print pause" command (^P^C) to replace the wheel for the single accent. The "print pause" is a toggle, so

remember to insert it directly before and directly after the accent. Clearly this method is desirable only on very short or selected documents because changing the wheel is very time consuming.

## III. "FIND AND REPLACE" TO INSERT DIACRITICS

### A. RATIONALE:

Experience has shown that unless you are working with a very short document (i.e. two pages or shorter), it is easier to insert the diacritics after the document has been completed. There are a number of reasons for this, but simple speed and accuracy are the overriding considerations. It is much faster to type and edit without the diacritics slowing you down and cluttering up the screen. Editing also opens the possibility of accidentally removing some of the print commands, especially since you would be most likely to edit with the print commands off the screen (^O^D). The solution is easy and efficient, far more than the old method of inserting the diacritics by hand: simply use the "find and replace" to insert the diacritics.

### B. "FIND AND REPLACE" COMMANDS:

The command for "find and replace" is **^Q^A. Wordstar** will provide prompts for the entire process, starting with the question "Find?" After typing the word or phrase, the program will respond with "Replace With?" At this point you type the word with complete diacritics and any other print commands (eg. boldface). After entering the new correct spelling with diacritics, you will have to choose the method of searching and replacement, with the prompt, "Options (? for Info)" The options (B,W,U,N,G) are important and should be completely understood.

1. **B = Search Backwards** - this does exactly what it says. The program will only search in one direction, so if you are in the middle of a document remember that it will only search in one direction from that point. The easiest method to avoid missing any words is to pass through the document starting at the beginning for one word and search backwards from the end of the document for the next word.

2. **W = Whole Words Only** - This option will only replace the word or phrase if it stands alone. Consequently, words that may appear in compounds (eg. Kṛṣṇa in Kṛṣṇadāsa) will not be changed if one resorts to this command. If you ignore this option, all appearances of that sequence of letters will be replaced.

3. **U = Ignore Case** - This option will select any sequence of characters that matches the "Find?" entry regardless of case, but will

replace it exactly as typed in the "Replace With?" entry. So if you have capitalized a word, you should note it and change it back manually or you should ignore this option and make a separate pass with the capitalized spelling. The latter is more efficient since it tends to eliminate the human error.

4. **N = Replace Without Asking** - By using this option the program will make the change without prompting; otherwise you will have to instruct the program from the **Y/N** (Yes/No) prompt, which will appear at the top of the screen.

5. **G = Replace in Entire File** - If this option is not used, the program will search for the first appearance of the word in question and stop. This command can speed up the process considerably, especially in longer files, and it avoids the need to retype the ^Q^A command to initiate the whole process again.

**C. HINTS AND CAUTIONS:**

1. Because of the system of transliteration and the inability of the machine to distinguish on the level of semantics, some words will just have to be changed manually (for example, the pair rasa/rāsa or bhava/ bhāva). In order not to miss any occurence, use the "Find and Replace" (^Q^A) command and make individual decisions as they occur, answering Y/N for each case (clearly you would not use the 'N' or "Replace Without Asking" option, but would use the 'G' or "Replace in Entire File" option).

2. Those plurals which are formed by adding 's' can be handled by ignoring the 'W' option (Whole Words Only) in "Find and Replace" (^Q^A). Otherwise, you will have to make separate passes through the document for the singular and plural forms.

3. If you misspell a word initially, the machine will not pick it up when it searches to replace diacritic marked words. If you are not an accurate typist, then the advantages of "Find and Replace" are limited. A dictionary or spelling checker program, especially one which allows you to add your own words (like **Spellstar®**), would be useful.

4. If you use shorthand while composing your text, the "Find and Replace" command will easily change your shorthand into the final full form of the word or phrase (eg. typing 'V.' for Vaiṣnava). Two cautions are in order. First, always be consistent in your abbreviations, or it will have the same effect as misspelling a word. Second, documents with manual footnoting may require extensive reformatting in order to accomodate the lengthened text created when the abbreviations are filled out. In longer documents with footnotes, I recommend taking the time to type

out the full word. However, inaccurate typists may find this an easy way to avoid typos and subsequently avoid mistakes while inserting diacritics. A footnote program will substantially reduce the problems created by the use of abbreviations or shorthand. Unfortunately, **Wordstar** does not have macros which would allow you to insert frequently used words or strings of words at a single keystroke; abbreviating and using the "Find and Replace" command is similar, but clumsier.

5. Never forget how literal the machine is. There is no substitute for accurate proofreading, regardless of the many advantages the word-processing software affords the user.

## IV. USER DEFINABLE FUNCTION KEYS

On many of the newer sixteen bit micros, such as the popular personal computers from IBM, DEC, AT&T, Compac, etc. you can reprogram some of the user definable function keys to simplify the insertion of diacritics. The changes can be made when installing **Wordstar** the first time or in a subsequent reinstallation. Because the function keys have already been assigned values, you should consider very carefully which keys are to be used. There are some inherent limitations due to the different hardware architectures and how effectively **Wordstar** has been adapted to the machine in question. For example, on the **DEC Rainbow**™ the allowable string of characters is limited to four for any given function key. Consequently, to program for the superscript or for the subscript, two keys are needed (eg. superscript macron: key 1 = ^P^T-; key 2 = ^P^T^P^H), which eliminates five keystrokes. For more details on how to reprogram function keys in the advanced versions of **Wordstar**, consult the installation section of the manual.

## POSTSCRIPT

Although **Wordstar** is an old program, MicroPro has successfully adapted it to any number of different micro-computers while improving its overall performance. As mentioned above, the program has limitations which are occasionally an annoyance, but overall, the technology allows the user to manipulate the text to suit his needs, rather than forcing the user to change his needs to fit the technology. Hopefully, in the future, the programs will allow even greater flexibility (eg. true proportional spacing using diacritics, etc.), providing us with the capacity to generate more professional documents without the fuss while substantially cutting the cost of printing.

* * *

# THE UNIVERSAL TYPEWRITER

by

David K. Wyatt [1]

and

Douglas S. Wyatt [2]

Imagine a typewriter on which it might be possible to type any alphabet. Or, even better, a polylingual and multialphabetic word-processor. As one who works with a variety of Southeast Asian alphabets I have long dreamed of such, but until recently could get no further than to procure Thai typing elements for my antiquated IBM Selectric. Like many, I drooled over the Xerox multilingual word-processor described in the July 1984 issue of Scientific American, but struck it from my Christmas list on realizing the cost of that machine.

However, if one is willing to put up with some relatively minor inconveniences, a very serviceable universal typewriter and word-processor already is in existence, as will become apparent from some of the sample printouts reproduced below. With the aid of master programmer Douglas Wyatt, we now have running a CP/M

---

1. Department of History, Cornell University, McGraw Hall, Ithaca, NY 14853.

2. New York Dairy Herd Improvement Cooperative, Ithaca, NY 14850.

microcomputer capable of both displaying and printing any alphabetic script that runs from left to right, whether or not it has complex superscripts and subscripts, and whether or not that script has ever even been printed before.

Our hardware is one of the least well-known, "plain vanilla" microcomputers on the market (and also one of the cheapest for the features it has): the Lobo MAX-80.[3] Quite apart from its ability to run thousands of existing software programs, and its capacity to accommodate up to eight floppy and/or hard disk drives, it is one of the rare machines around that does not store its displayable character set -- the alphabets and symbols one sees on the video display -- in permanent memory (ROM), but instead loads its character set into user-accessible memory (RAM) each time one starts up the machine.[4] It is therefore possible to change that character set in any way one wishes, including to non-Roman alphabets.

Furthermore, like most microcomputers the Lobo MAX-80

---

3. Lobo Systems, Inc., 318 East Gutierrez, P.O. Box 4626, Santa Barbara, California 93101; (805) 564-3356. The basic unit is priced at $945, plus monitor ($139) and disk drives (2 5" drives for $485; 2 SSDD 8" drives for $1185; 2 DSDD 8" drives for $1485). The basic unit has 128K RAM, running under CP/M 3.0; and it will also run the TRS-80-compatible LDOS.

4. There are several other microcomputers that, to my knowledge, handle their character sets similarly, including the Tandy 2000. Are there others?

regularly holds two separate character sets in memory -- normally a "bright" set and a "dim" set, or a regular set and one in inverse video -- the first green-on-black and the second black-on-green. Lobo allows the user to toggle between these two sets using any special function key s/he may designate, just as Lobo also allows one to redefine any key on the keyboard (for example, to create a Dvorak keyboard in place of the usual QWERTY one).

Just within the common realm of roman alphabets, the four users of this machine each employs his own, custom-designed character set; and James Wyatt sometimes will run the machine displaying Olde English or even Tolkien-ish runes, while ten minutes later Andrew Wyatt will be displaying a sans-serif font. Because I sometimes try to create bibliographical files that emulate OCLC records, I have redefined the backslash ("\") as a double-dagger on my display. Even this degree of flexibility is both unexpected and welcome in the microcomputer world.

But after a good deal of experimentation we have gone several steps further, to the point where I can now intermix roman with Thai or Lao scripts on the display screen. Recently, when transcribing an old Lao manuscript on the computer I encountered subscripts for which I had not provided when creating the Lao character set. I was able to add them to the display screen and to the printer in less

than an hour.

Let me describe briefly how this system works, by taking the reader through the sequence of actions I have to take to turn an "ordinary" microcomputer into a "universal typewriter."

The first requisite is to define a character set for display. The Lobo MAX-80's character sets are composed of dots (pixels) in an 8 x 8 grid. A public-domain BASIC program for defining characters for the MAX-80 is available from the Lobo users' group, MAXIMUL; but this is relatively slow, crude, and inflexible. Douglas Wyatt instead has created a character font editor, EDFONT, written in the C programming language (and therefore translatable for other machines), that allows for the creation and editing of character sets to any specifications, very conveniently and easily.[5]

The MAX-80's display character set is created and stored in groups of eight numerals for each letter in the set; each numeral defining the "lit" and "unlit" pixels in a single horizontal row. Thus the decimal numeral 131 (hexadecimal 83) when converted to binary form would represent a row of pixels in which the first, seventh, and

---

5. Commercially-available software from PowerSOFT, running under LDOS, also accomplishes these same character-set editing functions. We have not had occasion to test it.

eighth pixels were lit. Unclear? Decimal 131 in binary is 10000011 -- representing *.....** in pixels! This will become even clearer if we take the representation of a complete letter of the alphabet -- let us say the first letter of the Thai alphabet:

| | Binary | Decimal |
|---|---|---|
| ........ | 00000000 | 0 |
| .*****.. | 01111100 | 124 |
| *.....*. | 10000010 | 130 |
| .*....*. | 01000010 | 66 |
| .*....*. | 01000010 | 66 |
| .*....*. | 01000010 | 66 |
| .*....*. | 01000010 | 66 |
| .*....*. | 01000010 | 66 |

Thus the Thai "k" character will be represented as 0,124,130,66,66,66,66,66 for display purposes.

To print the same character, printers ordinarily expect the pixels of a character to be defined in terms of their vertical, rather than horizontal, alignment. For printing purposes, therefore, the Thai character above (reading the binary numerals vertically rather than horizontally) would be 32,95,64,64,64,64,63,0. Douglas Wyatt also has written utilities to convert horizontal display characters into vertical printing characters. For a sample, Figure 1 below reproduces this paragraph printed in James Wyatt's Runes font, created on an 8 x 8 grid for display and converted to the Epson FX-80's 9 x 11 printing characters.

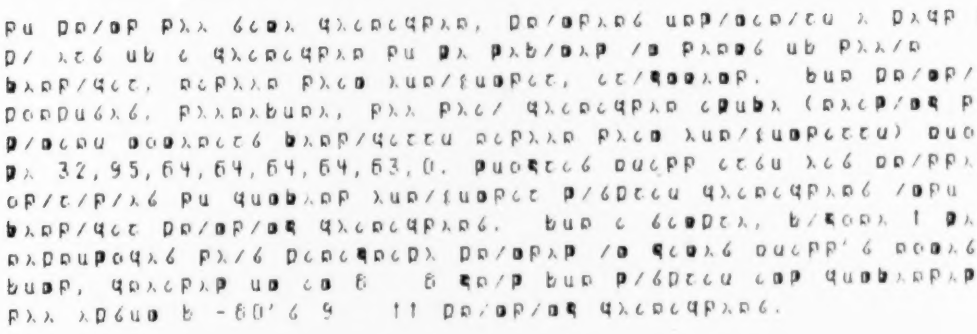

Figure 1. -- The paragraph above, printed in James Wyatt's runes font.

Upon starting up, the MAX-80 displays its standard roman character set on the screen -- the usual green-on-black, with an inverse-video black-on-green character set accessible by toggling Function Key 4 (which can be defined for that purpose). If I wish, let us say, to work on entering a Lao chronicle text into the machine, the first thing I do is to change my character set. I run a program written by Douglas Wyatt in "C" that loads the character set I have called "LAO" into memory. This operation retains the usual roman character set in regular video, but replaces the inverse-roman set with a regular-video Lao character set. When this is completed, in about two seconds, I habitually (and unnecessarily) test the result by toggling the F4 key and typing a few words on the screen, which are displayed in Lao. I then toggle the F3 key to restore roman characters. I then enter my word-processing program (Perfect Writer) as normal; and,

when it is ready, simply toggle the F4 key again to activate the Lao character set. That's all there is to it: Anything I type from that point on is in Lao. Should I need to read Perfect Writer's status messages on the bottom line of my screen, I simply toggle back into roman script for a moment, then return to Lao.

And here we get to two significant deficiencies of the system. First, and perfectly obviously, it is not possible to have all possible alphabets engraved on the surface of every key. One either learns Lao (or Thai, or whatever) touch-typing very quickly, or one relies on wall-charts. I have found that it makes good sense to maintain alphabetic correspondences between alphabets to the greatest extent possible: thus, the "k" keys of both Thai and Lao are on the roman "d", the "long-a" vowel of both is on roman "k", and so forth. Second, a much more serious deficiency occurs in the cases of all languages with superscripts and subscripts, which includes all Southeast Asian scripts of Indic origins. It is possible to display these properly placed above or below the consonants, but the programming tasks involved seemed so immense that we have not thought it worth the effort. What we have, therefore, is a system in which letters are displayed exactly as, and in the order as, they are typed. (A sample of a Lao input file is given in Figure 2.)

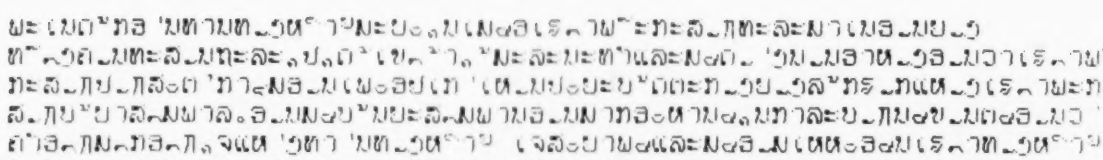

Figure 2. -- Sample of Lao-language input file, with superscripts and subscripts all included on the line, approximately as it would be displayed on the video screen.

This takes a great deal less getting used to than I would have expected; and I should get ahead of myself a moment to note that this deficiency is a feature primarily of the text-entry stage alone. It also means that, because superscripts and subscripts are mixed into the main line of consonants, it is difficult to aim for, let alone to achieve, printed lines that are justified at both margins.

Having completed entering my Lao text, I exit from the word-processing system and go into Microsoft BASIC for the purpose, first, of converting a monolinear Lao text in which super- and subscripts are mixed in with the consonants into a multilinear text in which the super- and subscripts are properly placed. This is basically a simple filter program which takes my original Lao manuscript input file (which has the file-name extension .LMS, for Lao ManuScript) and goes through it character-by-character. For each line of input file, it creates four lines of an output file: line 1

contains all superscript tone markers, line 2 contains all superscript vowels, line 3 contains the on-the-line consonants and vowels, and line 4 contains all subscripts. Each character is placed into the appropriate line of the output file, which is given the file-name extension .LAO. The resulting file, if it were to be displayed, would look like Figure 3.
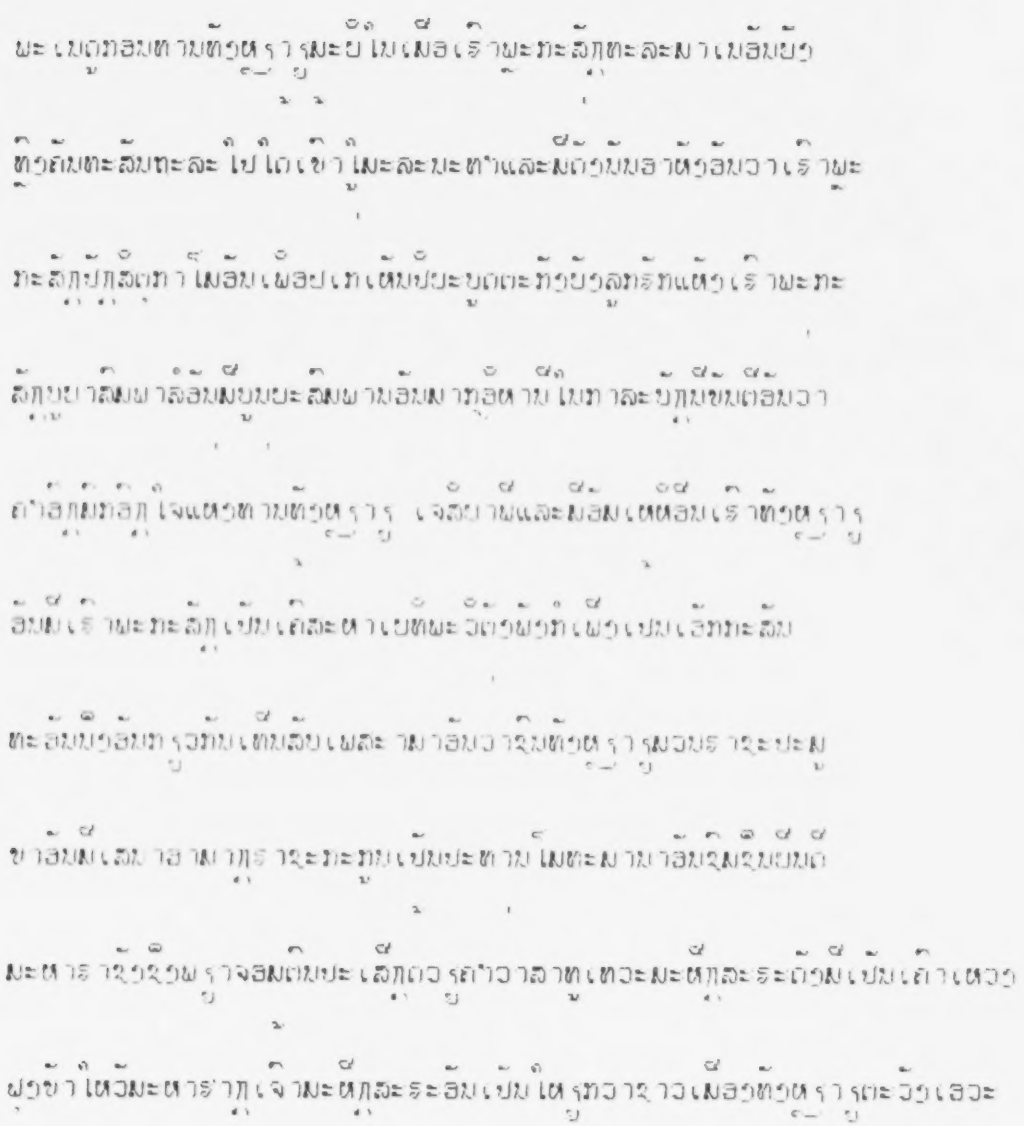
Figure 3. -- Lao output file, showing how the input lines given in Figure 2 have been broken out into four lines, containing tone markers, superscript vowels, consonants, and subscript vowels, respectively.

If I wish to proceed on to print the file, I first load a Lao character set into my printer, in this case an Epson FX-80, which has the capability of accepting character sets downloaded from the host computer. (We also have successfully run non-roman character sets as graphics on the Epson MX-80, but that is much slower and, on the whole, less satisfactory.) A simple BASIC program accomplishes the font change.

Finally, another BASIC program handles the tricky business of printing Lao, basically by making four passes across the page to print each line originally input. The first pass prints all tone markers for a line. Without advancing the paper, the print head then again passes over the same space, adding superscript vowels. The paper then advances before the consonants are printed below the superscripts, and then advances slightly again before printing the subscripts. The final product of this process comes quickly from the printer looking like Figure 4, reproduced at the end of this article.

In the final printing, of course, the quality of the sample can be manipulated much as roman type can, using the emphasized, double-strike, condensed, and other print features of the printer. For most purposes, this should be more than adequate; and at the very least it often will allow the printing of scripts that otherwise could be

reproduced only in manuscript hand.

Obviously the "universal typewriter" system described here is to some degree hardware-dependent; but it need not be. Many microcomputers can be tricked into looking for their character sets in RAM memory, and thus can accept user-defined character sets; and even printers that cannot accept downloadable fonts usually will allow the printing of alternate character sets using their graphics mode. I, for one, however, would be hard pressed to think of another computer-printer combination that might be more congenial to the needs of those using non-roman alphabets than the Lobo MAX-80 and the Epson FX-80.

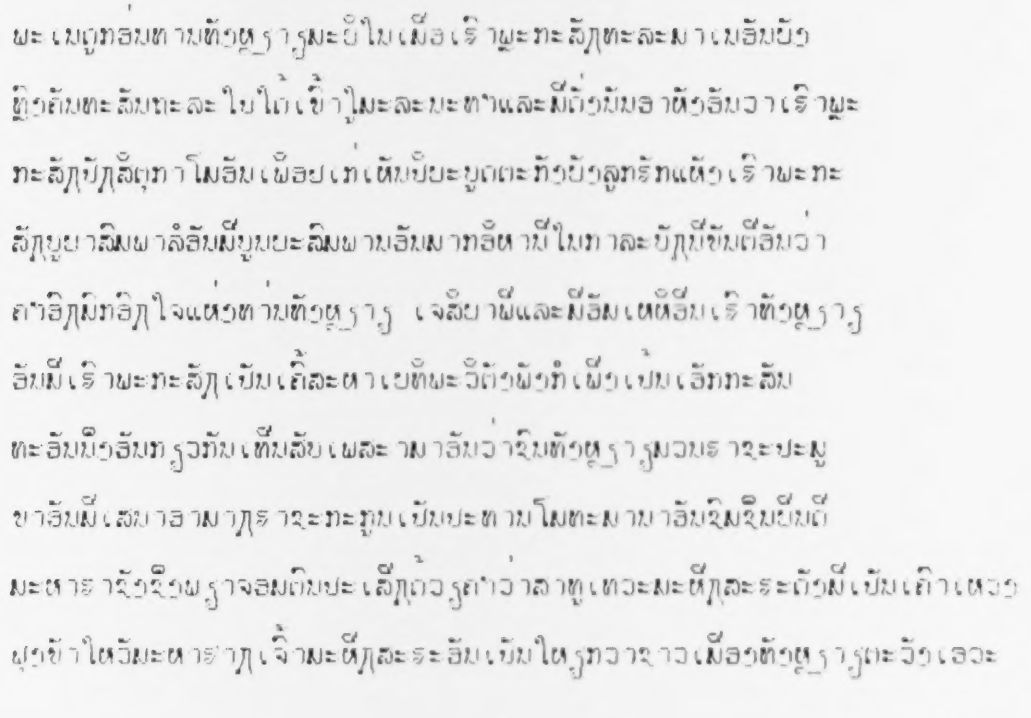

Figure 4. -- Final printing of portions of the file shown in Figures 2 and 3, demonstrating the collapse of vertical spacing.

# Multilingual Word-Processing Systems:

## Desirable Features from a Linguist's Point of View

by Lloyd B. Anderson

Ecological Linguistics
316 A Street, S.E.
Washington, DC 20003

All writing systems in a single flexible program?

The technology is now becoming affordable so that microcomputers can in principle handle all written characters and writing systems in forms almost as good as commercial typesetting. The software programming lags behind, however, and each new typefont and script is still programmed individually at an enormous cost in time.

It would be nice if we could have a single software program which is flexible enough to handle all written languages within a single word-processing environment, and which can easily add a new typefont or writing system which a user wants. What follows is one attempt to list the "module" program components needed. You are encouraged to make your own additions, to use such a scheme in asking manufacturers how far they can meet multilingual needs, and to send your critiques to the author, who will make it known when substantive updates are available.

Convenience for the user:

The user should only have to specify from the keyboard which language is desired and if appropriate, specify a less common type font or method of keyboard input. The program should automatically do the following without further instruction:

a. *Choose the proper character set* (with its dot-matrix adaptations to particular sizes, screens, printers, and so on).

b. *Remap the keyboard* to fit the new character set and the standard keyboard for that language. If desired, the normal case could be to allow input via transliteration from Latin-letter or other keyboards, where sometimes two or more keystrokes will identify a single character, as sh giving Russian ш.

c. *Specify alphabetical order* (sorting order) of the characters which will govern all indexing programs (unless for bibliographies and library catalogs, where different writing systems and their transliterations are to be mixed, but following the single alphabetical order of the receiving language).

d. *Handle the physical arrangement of base characters and diacritics, accents and vowelings.* This includes the order which might be left-to-right, right-to-left or top-to-bottom. It includes the accent marks above letters in European alphabets, which the program must raise to higher levels above capitals and letters with vertical extenders. Where this is not done, as on most typewriters and printers, we see less than ideal results as over the capital letter E following: é is ok, but É is not so good.

Fuller flexibility is needed for vowels in Semitic scripts and in India and Southeast Asia. Here a single vowel may be represented by components to the left of a consonant, above, below or to the right of a consonant, or by a combination of several of these at once. Thai adds tone marks as well, giving multiple diacritics above a single letter.

Letters may vary their form according to context, as in many "conjunct" characters of Indic alphabets, or the linked letters of Arabic. This too should be automatic.

Linear order of pronunciation, typing and code transmission may not correspond to linear order of print. An example of this is in Hindi, where the vowel /i/ is written before a cluster of consonant letters even though it is pronounced after them.

All of these should be independent modular concepts. For the next generation of word-processing programs, there should be a standard "interface" using concepts like "line of text" independent of the font size or direction of writing, and separate concepts for number-of-characters-per-line and number-of-lines-per-page; using a concept "above-letter-accent" which does not specify exact position, and a separate concept of "extender-height" which is used to calculate how high to place an accent (or a set of categories for base characters of different heights matched to a small list of accent heights).

The user's choice should be as simple as pressing **Home** or **Alternate** keys to switch between two previously designated writing systems, fonts, or the like, and a simple way to specify in advance just which writing system or font is referred to by each of such keys.

Where are we now?

Keyboard-remapping programs are now available (separately, as Prokey, or packaged as part of larger programs). The keyboards we want are not already prepackaged, of course, so we must construct them.

Large character sets and exotic alphabets are available from some special suppliers. Unfortunately, it is not yet generally possible to integrate these with off-the-shelf word-processing or database programs, partly because most of these are restricted to 7-bit ASCII codes and allow only 96 characters, not permitting the additional 96 character codes for printing characters which are available in an 8-bit ASCII code.

In most programs for exotic languages, each combination of base character with an accent or other diacritic must be separately specified as a complete character. As graphics capabilities become more general (we're just on the edge of it now), the accent marks can be placed on a separate "layer" which is merged with the base character in real time on the screen or in printing, but remains separate in programming. Some programs already permit this.

With more complicated alphabets and with Chinese characters, a much higher screen resolution is needed so they can be distinguished from each other. A compromise does now exist, in that printer detail can be of much higher resolution than what is seen on the screen under certain programs (this is true for printers with "graphics" capabilities). At low screen resolutions, only a small part of a page could be viewed at any one time if the characters are to be distinguished. Screens with much higher resolution (1024 by 1024 pixels (dots) instead of the 320 by 200 of the IBM PC, for example) are now becoming available, but the software "drivers" required to operate these screens cannot work with most of the existing packages of software. So far, they are all intended mainly for people doing their own "graphics" programming, which excludes almost all linguists.

Size variation is not a trivial matter. Simply multiplying each pixel by two or four either vertically or horizontally or both can indeed produce oversized

characters, but they have a "stairstep" form, reflecting the discrete dots in their origins. A more sophisticated "zoom" is available in some programs, which operates only a highly detailed space mapped inside the computer, by dropping out pixels as one zooms back from the image and it becomes smaller (and correspondingly a larger fraction of the page in the computer's memory can fit on the screen).

Typefont variation is extremely expensive to include at present, since each dot of each dot matrix must be specified and that requires a lot of memory. There are other methods of specifying letters, in terms of pen-strokes or brush-stroke shapes. See "Digital Typesetting" by Charles Bigelow in **Scientific American**, August 1983, especially pages 118 to 119. By factoring out separate variables in the type style such as stroke width or slant or presence versus absence of serifs, fonts can be specified in more flexible ways. This requires an extensive program to decode the parameters and build the characters "on the fly", each time they are used. Such programming has been developed for Latin letters by Donald Knuth in the TEX system, and has even been applied to Chinese characters. In this way, a larger sized character is derived from the same formula as the smaller one, and is equally smooth in design since the formulas generate a smooth sequence of dots no matter what the character size.

Typesetting is also on the edge of becoming cheaply and widely available, and a number of programs exist for this. Some are very sophisticated, such as Donald Knuth's system mentioned above, but do not permit interactive editing.

The above notes are intended as an aid to help you when inquiring about what a program can do for you.

## Reviews of Organizations, Books, Journals and Articles

In this section, we will regularly review all resources of interest except hardware and software, which will be covered in another section. If you know of any books, journals or articles which would be useful to others working with Asian and Middle Eastern languages on computers, please send information about them to the editor. If it's an article, a copy of it would be appreciated. We also want to mention any organizations of interest, which is where we'll begin this issue's column.

## Organizations

There are several organizations which will be of interest to at least some readers. The **Association for Literary and Linguistic Computing** has existed for well over a decade now. Its primary aims are to "further literary and linguistic research by computer and to provide a means of communication for all those concerned with such research". The **ALLC** has two publications: the first is the **ALLC Journal**, a biannual refereed journal carrying articles on the results of literary and linguistic computing. The second is the **ALLC Bulletin**, published three times a year. It is less formal and carries articles on methodology, software and work in progress. Membership for individuals in the **ALLC** costs £10 and includes both the **Journal** and the **Bulletin**. For further information contact: Dr. T. Corns, University College of North Wales, Bangor, Gwynedd, LL57 2DG, United Kingdom.

While most of their members work with European languages, there are two specialist groups of interest to our readers. One specializes in Hebrew and is managed by Professor Y.T.Radday of the Technion in Haifa, Israel. The other, formed at the last annual conference of the **ALLC**, specializes in Languages and Literatures of the Indian Subcontinent. It is chaired by Dr. Peter Schriener of the Seminar für Indologie und Vergleichende Religionswissenschaft, Universität Tübingen, Münzgasse 30, 7400 Tübingen, West Germany. Please see the Announcements Section in this issue regarding the status of this group. The **ALLC** will be of interest to anyone involved with using computers in a scholarly context. It has a broad international base of members, with many from Europe.

The **Association for Computers and the Humanities** was organized to encourage "computer aided research in language and literary studies, history, anthropology and related social sciences as well as the use of computers in the creation and study of art, music and dance. In addition, the Association fosters the development of computer technology and techniques necessary for these studies". Benefits of membership include a quarterly newsletter covering activities of the Association and other topics, the option of a reduced subscription rate to the journal **Computers and the Humanities** and reduced registration fees for meetings of the ALLC and the International Conference of Computers and the Humanities which the **ACH** organizes. Annual membership costs $15. For more information, contact: Harry Lincoln, Association for Computers and the Humanities, Music Department, SUNY Binghamton, NY 13901. The **ACH**, like the **ALLC**, will be of interest to anyone using computers in a scholarly context.

Its membership is predominately American, but there are members from the rest of the world as well.

The **Chinese Language Computer Society** was formed in 1976 with the goal of advancing "the science and technology of information processing in Chinese and other languages containing a non-trivial Chinese component" and promoting "the exchange of information on Chinese language information processing in the best scientific and professional tradition". The languages that members of the **CLCS** are primarily interested in are Chinese, Japanese and Korean. This society has a computer science rather than humanities orientation. The benefits of the **CLCS** are a quarterly journal entitled **Computer Processing of Chinese and Oriental Languages**, a newsletter and international conferences (papers presented at them are usually available in Proceedings). Annual membership costs $15. For more information, contact Professor Helena Gin Wong, CLCS Membership Chairman, Computer Science Dept., San Francisco State University, 1600 Holloway Ave., San Francisco, CA 94132 or telephone (415)469-2858 or 469-1008. The **CLCS** is oriented towards computer scientists and programmers, but there are articles published in their journal that can be read by anyone with a basic familiarity with computers.

The **Computer Society of India** is the national computer society in India and corresponds to the Association for Computing Machinery in the United States. Benefits include a newsletter, the **CSI Communications**, published monthly and a journal, **Computer Science and Informatics**, published twice a year. Both are published in English. Membership inside India costs Rs. 60/- and outside India it costs $30 (which includes the cost of airmailing all publications). For further information, contact the Computer Society of India, Institution of Engineers Building, 15 Haji Ali Park, Bombay 400 034, India. The **CSI** is a professional society oriented towards programmers and computer scientists. There are only occasional articles on Indian languages on computer.

The **Association for Computational Linguistics** was founded for persons interested in using computers for language research (including phonetics, phonology, syntax, semantics, discourse, ...), applications (including translation, office automation, lexicography, ...) and scholarly investigations (including stylistics, content analysis, ...). Benefits include a newsletter and a journal, **Computational Linguistics**, published quarterly, and an annual meeting. Annual membership costs $20 (I think). For more information, contact Donald E.Walker, ACL, Bell Communications Research, 445 South Street, Morristown, NJ 07960 or telephone (201)582-7406. The **ACL** is a specialist organization that seems to appeal primarily to computer scientists, especially those working with natural languages and artificial intelligence. The articles in their journal are sophisticated and can generally only be read by advanced linguists and computer scientists.

## Books

Not very surprisingly, there are few books on Asian and Middle Eastern languages on computer. There are several books that are not directly relevant, but still useful. If you know of any others, please let the editor know of them.

**A Guide to Computer Applications in the Humanities**, by Susan Hockey (Johns Hopkins University Press, Baltimore, 1980) is an excellent introductory work that covers much of what has been done in the last two decades in literary and linguistic computing. There are only a few references to any work with non-western scripts, but on the other hand it does cover a large range of which sorts of things have been done with European languages. The chapter titles are: Introducing Computers; Input and Output; Word Indexes, Concordances and Dictionaries; Vocabulary Studies, Collocations and Dialectology; Morphological and Syntactic Analysis, Machine Translation; Stylistic Analysis and Authorship Studies; Textual Criticism; Sound Patterns; Indexing, Cataloguing and Information Retrieval; and How to Start a Project. Also included is a useful bibliography. There have been more recent developments that make some of the material a little dated, but it is an excellent place to start for anyone interested in or involved with literary and linguistic computing.

**Text Processing: Algorithms, Languages and Applications**, by Allen B. Tucker, Jr. (Academic Press, New York, 1979) is a somewhat dated but still useful introductory work on literary and linguistic computing. The chapter titles are: Introduction to Text Processing; Introduction to PL/I for Text Processing; Introduction to SNOBOL for Text Processing; Overview of Text Processing Packages and Applications; and Literature Review. This book would be most helpful to someone who is interested in literary and linguistic computing, but does not know much about computers. The articles and books cited were all published before 1979.

**TEX and METAFONT: New Directions in Typesetting**, by Donald Knuth (Digital Press, Bedford, Mass., 1979) is already a classic. It describes a typesetting system (called TEX) and typeface designing system (called METAFONT). Both were conceived of, designed and (primarily) implemented by Donald Knuth, who has also been writing the series **The Art of Computer Programming**. Both systems are batch oriented and probably cannot be made interactive (some claim for good reason too). Even so, there are many concepts in these systems which have changed the way most people look at typesetting and typeface design. This book explains the concepts behind the two systems. TEX has been used to typeset well over a dozen books and is used to typeset lots of mathematics, one of the most difficult tasks you can give a typesetter. Some work has been done with Indian languages and Chinese, but much remains to be done to make TEX usable for Asian or Middle Eastern languages.

Another recent book about these systems, also by Donald Knuth, is **The TEXbook** (Addison-Wesley, Reading, Mass., 1984). This book explains how to use TEX for both beginners and advanced users. It is easy to read and is even humorous in

parts. I should add that TEX and METAFONT are in the public domain. For more information contact the TEX Users Group, c/o American Mathematical Society, PO Box 1571, Annex Station, Providence, RI 02901. The TEX system is now quite common on university computer systems in the United States and Europe, especially those with strong mathematics departments. The code for the TEX system is available for a nominal fee from the TEX Users Group.

**Coded Character Sets, History and Development**, by Charles Mackenzie (Addison-Wesley, Reading, Mass., 1980) is (contrary to its title) primarily just a history of the IBM EBCDIC (Extended Binary Coded Decimal Interchange Code) and the ASCII (American Standard Code for Information Interchange) character codes used to represent characters in computers. It does not cover any other character codes at all, nor are there any references to other books or articles. This is frustrating because there are so few in existence. It does explain in great detail why the EBCDIC and ASCII codes are structured the way they are and some of the problems one runs into when converting from one to the other. It would be useful to anyone designing computer character sets as a source of background information.

**Word Processing Handbook**, by Ivan Flores (Van Nostrand Reinhold, New York, 1983) describes in great detail the components of a word processing system, both software and hardware, the various features included in almost all word processors and a chapter on how to evaluate word processing (WP) systems. This book would be useful to anyone designing or implementing a word processing system because it lists and explains the features of a WP system in more detail than I've ever seen anywhere else. Features are described as a user would see them, not from the viewpoint of a programmer or designer. Features of the most popular word processors are described too, giving a feel for what is popular (and presumably comfortable) for most people. There is no discussion of any languages other than English and no references to any other books or articles.

## Journals

Several journals have been described in the section on organizations. Their names once again are:

- **ALLC Bulletin** and **ALLC Journal**, both published by the Association for Literary and Linguistic Computing
- **Computer Processing of Chinese and Oriental Languages**, published by the Chinese Language Computer Society
- **Communications of the Computer Society of India** and **Computer Science and Informatics**, both published by the Computer Society of India
- **Computational Linguistics** (formerly **American Journal of Computational Linguistics**

In addition to these journals, there are several others of interest.

**Computers and the Humanities** has been published for over fifteen years now. While many of the articles are only of peripheral interest, there are still several articles published each year that directly concern Asian or Middle

Eastern languages on computer. Many of the other articles will be of interest to anyone doing literary or linguistic computing. In addition to articles, they also regularly review books and software as well as publish a list of scholars active in the field. By joining the Association for Computing in the Humanities, you can receive a discounted subscription to it. For more information about the journal, contact the publishers: Paradigm Press, Post Office Box 1057, Osprey, FL 33559, or the **ACH**.

**SCOPE (Scholarly Communication: Online Publishing and Education)** is a newsletter which is also published by Paradigm Press. It comes out every two months and carries news on new books and journals of interest to humanities and social science users of computers. Topics covered include interesting hardware, software and databases and upcoming meetings. It complements **C&H** rather nicely. Subscriptions are $47 per year and a reduced rate is available to members of the Association for Computing and the Humanities. For more information, contact: SCOPE, Paradigm Press, Post Office Box 1057, Osprey, FL 33559, or the **ACH**.

**Technical Japanese Translation** is a unique newsletter published by Dr. Donald Philippi. Its primary audience is technical translators of Japanese, but anyone interested in Japanese on computers should also subscribe to it. It covers developments using the Japanese language on software and hardware from a user's point of view in addition to topics primarily of interest to technical translators such as word lists, reviews of dictionaries and other books. A six month subscription costs $20 in the US. Subscribers in other countries should contact the editor. The first year's back issues (numbers 1 through 15) cost $20 in the US and $25 for Japan and Europe. For further information, contact the editor: Donald Philippi, 715 Tenth Ave., San Francisco, CA 94118. His telephone number is (415)752-7735.

The **Seybold Report on Word Processing** is a specialized and expensive newsletter that reviews word processing systems in great and exhaustive detail. So far as I know, they have not reviewed many WP systems that work with Asian or Middle Eastern languages, but the reviews are the best I have ever seen. If anyone is contemplating buying, using or designing a system or piece of software for word processing, this newsletter cannot be too highly recommended. There are several other **Reports** of potential interest, including the **Seybold Report on Office Systems** ($60/year) and the **Seybold Report on Professional Computing** ($120/year). For more information, contact: Seybold Publications, Box 644, Media, PA 19063. Their telephone number is (215)565-2480.

The **Newsletter of the Society for the Study of the Indigenous Languages of the Americas** has a regular column for computer users. While the particulars may not be directly applicable to Asian or Middle Eastern languages, there are topics discussed that are still quite useful. For more information, contact the editor: Victor Golla, Department of Anthropology, George Washington University, Washington, D.C. 20052.

The **Small Computers in Libraries Newsletter** has occasional articles that deal with Middle Eastern (and hopefully Asian) languages in libraries. A one year subscription costs $20 in the USA, $25 in the rest of North America and $35 in the rest of the world. For more information, contact: SCIL, Graduate Library School, College of Education, University of Arizona, 1515 E. First Street, Tucson, AZ 85721. They can be reached by telephone at (602)621-3566.

## Articles

In the course of tracking down people who are working with Asian and Middle Eastern languages, several people have been kind enough to send me articles of interest. If you have written any articles of interest to the readers of this newsletter, please send a copy of the article to the editor (or at least a reference to it) so that others can be told of it.

James E. Agenbroad, "Character Sets: Current Status and East Asian Prospects", **Journal of Library Automation** 13(1):18-35. March 1980. Recent standards for East Asian character sets are described here, along with a description of their potential usability in libraries. Character sets for English (ASCII and others), Cyrillic and Greek are also illustrated.

Mohammed M. Aman, "Use of Arabic in Computerized Information Interchange", **Journal of the American Society for Information Science** 35(4):204-210. (1984). This article discusses the problems of putting Arabic on computer and the new Arabic character standard (CODAR-UFD). A list of vendors of Arabic terminals and computers is also given.

Joseph D. Becker, "Multilingual Word Processing", **Scientific American** 251(1): 96-107. (July 1984). The problems of developing a word processor that will work with most of the world's languages are described here. The article clearly describes why a truly multilingual word processor is so difficult to implement.

R. W. Bemer, "Inside ASCII - Parts I,II and III", **Interface Age**, May 1978: pp.96-102, June 1978: pp.64-74 and July 1978: pp.80-87. This series of three articles in a popular computer magazine explains the details of the ASCII character set, which has been slightly modified into International Standard 646. This character set is the basis of all national character sets and is therefore of interest to everyone. These articles, by one of the people involved in designing several character sets, also describe character sets for Cyrillic, Greek and Japanese.

Winand M. Callewaert, "The Leuven Text-Composition System in Devanagari", **Computer Science and Informatics**, 13(2): 29-30. (1984). This article describes a system used for Hindi, Sanskrit and Gurmukhi. The keyboards and character codes used for each of the languages are described.

Y. Choueka, M. Cohen, J. Dueck, A.S. Fraenkel and M. Slae, "Full Text Document Retrieval: Hebrew Legal Texts", **Proceedings of the Symposium on Information Storage and Retrieval**, April 1-2, 1971, pp 61-79. (Another ACM publication). An early paper describing the initial phase of the Responsa project. This project aims to store and make easily available an enormous quantity of Hebrew and Aramaic texts. The problems of dealing with Hebrew are well described.

International Business Machines Corporation, "IBM 5291 Display Station: National Language Requirements (Thai) Operator's Guide: RPQ 8D0025", **GA09-1665-0** (File No. S5250-14). This document describes the keyboard layout and graphics for a Thai version of their 5291 terminal.

J. B. Millar and M.C. Newey, "An Arbitrary Font Printing System", **Proceedings of the Digital Equipment Computer Users Society**, Townsville, Queensland,

Australia (August 1977). This article describes a system that can be used to print in arbitrary fonts, including Devanagari. The font editor is also described.

J. B. Millar and W.W. Glover, "Synthesis of the Devanagari Orthography", **International Journal of Man-Machine Studies**, 14:423-435 (1981). This article describes an application of the arbitrary font printing system described in the article previously mentioned. The system was used to prepare material in Nepali. The system implementation and algorithm for synthesizing characters are described.

J. B. Millar and H. Oasa, "Proposal for ASCII coded phonetic script", **Journal of the International Phonetic Association**, 11:62-74. (1981). This article proposes a character set for the international phonetic alphabet.

Ichiko Morita, "Japanese Character Input: Its State and Problems", **Journal of Library Automation** 14(1):6-23. March 1981. Descriptions of most, if not all methods for entering Japanese into a computer and their advantages and disadvantages are given. Problems of working with kanji inside a computer are also described.

R.M.K. Sinha and Arjun Raman, "A Modular Data Terminal for Indian Languages", **Computer Graphics** 14(1&2):39-72. July 1980. A detailed design of a terminal capable of working with all of the Indian languages is described. Different methods of encoding the languages are described as well. **Computer Graphics** is a publication of the Special Interest Group in Graphics of the Association for Computing Machinery (ACM), the national computer society of the USA.

There are short articles of some interest occasionally seen in **Infoworld**, a weekly magazine that covers the computer business and new products. Some recent articles along these lines included: "Japan on 16K a Day", pp. 66-68 (28 May 1984) and "Saudi Arabia Embraces Micros", pp. 25-26 (23 April 1984). The publisher's address is: Infoworld, Circulation Department, 375 Cochituate Road, PO Box 837, Framingham, MA 01701.

As this issue was being completed, a special issue of **Computer**, a monthly magazine produced by the Computer Society of the IEEE (Institute of Electrical and Electronic Engineers), has just come out on Chinese/Kanji Text and Data Processing. There are eight articles on this topic, including an article by Joseph Becker who wrote the **Scientific American** article mentioned above. Anyone interested in Chinese or Japanese on computer should get a copy of this issue.

## Calendar of Events

**February 25-28, 1985 - International Conference of the Chinese Language Computer Society** will be held in San Francisco, California at the Golden Gate Holiday Inn. For further information, contact: Dr. Daniel T. Chang, Local Arrangements Co-Chairman, A45/098, IBM Corp., 1126 Olive Branch Court, San Jose, CA 95120. (408)268-7873.

**June 22-24, 1985 - Fifth International Conference on Databases in the Humanities and Social Sciences** will be held at Grinnell College. Major topics will be opportunities for database users provided by microcomputers, videodiscs, communication networks, online catalogs, and electronic bulletin boards. The registration fee is $125. For further information, contact: F. Moberg, ICDBHSS/85 Coordinator, Grinnell College, PO Box 805, Grinnell, IA 50112. (515)236-2570.

**June 26-28, 1985 - 1985 International Conference on Computers and the Humanities** will be held in Provo, Utah at Brigham Young University. For further information, contact: Randall Jones, Humanities Research Center, 3060 JKHB, Brigham Young University, Provo, UT 84602.

## Announcements and Queries

**Dr. Peter Schreiner** is the chair of a new **specialist interest group in Languages and Literatures of the Indian Subcontinent**, which is one of many groups of the **Association for Literary and Linguistic Computing**. He has written that "during the initial phase I see my task mainly as that of gathering information about people, institutions, projects, etc., to be exchanged and handed on whenever I am approached.

"Such information may concern computer-assisted work on any Indian language or text of any period (Vedic, classical Sanskrit, Prakrit, Pali, modern languages, dialects, oral tradition, etc.), including standards, routines, programs, etc. for the transliteration of Indian scripts and/or the handling and output of texts in original scripts (e.g. Devanagari); and such work may be of any type (lexicography, concordances, language teaching, bibliography, etc.).

"If you are working in this field, kindly let me know the following details concerning your project:

- Names and addresses of people involved
- Title and brief description of the project
- List of publications (published and forthcoming) based on the project
- Information on hardware and software used
- Lists of texts available in machine-readable form indicating whether they are available for outsiders and under which conditions.

"Do not hesitate to comment on what you expect from this specialist group or to suggest how it should function - we are just beginning the work! Thank you!"

Dr. Schreiner can be contacted at the the following address:

Seminar für Indologie und Vergleichende Religionswissenschaft
Universität Tübingen
Münzgasse 30
7400 Tübingen
West Germany.

**Maurice J. Bauhahn** is interested in hearing from anyone working with **Khmer** or **Thai on computers**. He can be contacted at the following address:

Maurice J. Bauhahn
c/o World Vision Foundation of Thailand
G.P.O. Box 1717
Bangkok 10500
Thailand

# Newsletter for Asian and Middle Eastern Languages on Computer

Featuring articles and news about Asian and Middle Eastern Languages on computer

Volume 1, Number 2 September, 1985

---

**Editor's Page:**
Anthony Meadow ..... 1

**About the Authors:** ..... 2

**Correspondence:** ..... 3

**Articles:**

A Note on the Production of Macrons in Transliterated Japanese
Jay Rubin ..... 4

Indian Fonts on the Macintosh
George Hart ..... 9

**Regular Departments:**

**Reviews:**
Books ..... 14
Journals ..... 16
Articles ..... 17

**Products:**
Hardware ..... 21
Software and Services ..... 23

## Policies

The Newsletter for Asian and Middle Eastern Languages on Computer is published by Bear River Systems. The entire contents of this Newsletter are copyright (c) 1985 by Bear River Systems. For rights to reproduce the material in this newsletter in any form or media, please contact the editor.

A one year subscription to the newsletter costs $10 per year throughout the world and always includes the entire volume. Subscriptions from the USA help subsidize subscriptions to the rest of the world. Issues are mailed first class in the USA and airmail to the rest of the world. To order a subscription, please send a check or money order for $10 (in US funds) payable to "Bear River Systems" to the following address:

Bear River Systems
PO Box 1021
Berkeley, CA 94701 USA

Telephone: (415) 644-1738

The newsletter is produced on a part-time basis and there is not sufficient time to deal with various forms of red tape. Please therefore send a check or money order, rather than a purchase order. If there is no other way to order the newsletter than through a purchase order, then add $5 to cover the additional work required to process the paperwork.

Exchanges with other newsletters will be considered. Please contact the editor.

This newsletter was produced using an Apple Lisa (aka Mac XL), MacWrite, Microsoft Word and the Apple LaserWriter.

## Submitting Articles

To submit an article for publication, please send one copy of the article in "camera-ready" form to the editor at the following address:

Editor, Newsletter for Asian and Middle Eastern Languages on Computer
Center for South and Southeast Asian Studies
260 Stephens Hall
University of California
Berkeley, CA 94720 USA

Electronic submission of articles is encouraged. Please contact the editor regarding this.

## Editor's Page

Welcome to the second issue of the Newsletter for Asian and Middle Eastern Languages on Computer. In this issue we have two articles: Jay Rubin extends the procedures described by Tony Stewart in the previous issue to SSI's WordPerfect and Microsoft's Word. Both are well known word processing programs for the IBM PC and compatibles. George Hart has written an article about his work developing Indian fonts for the Apple Macintosh computer.

In this issue, I have begun listing software and hardware of potential interest to the readers of this newsletter. These are simply listings of information provided by the manufacturer or from some other source. They are not reviews. In general, I do not have access to enough machines to try many of these packages, nor the time. If some of you have used any of these or other products, many others would be interested in hearing of your experiences. A review could be as short as a paragraph or as long as several pages. Send your reviews to the editor at the address given on the inside cover. I will publish articles written by developers of hardware and software products, and they will be labeled as such. If you are interested in writing such an article, please contact me.

All reviews and product listings were written by the editor.

A number of people greatly helped in getting this issue of the Newsletter out. I would like to thank Barbara Guerlin and Bruce Pray of the Center for South and Southeast Asian Studies at the University of California, James Agenbroad of the Library of Congress, James Nye of the University of Chicago and Ken Logan of the South and Southeast Asian Library at UC Berkeley for helping with ideas, and information about articles and books. George Hart of the Department of South and Southeast Asian Studies at UC Berkeley has been very helpful in many ways, including sharing ideas and equipment, such as the LaserWriter that this newsletter was printed on. Many thanks also to Tony Stewart at the University of Chicago and Maurice Bauhahn working for the World Vision Foundation in Thailand for letting many other people know about this newsletter. Lastly, I would like to thank my wife Diana for all her assistance in all aspects of publishing this newsletter.

This issue has come much later than I thought it would. My apologies to one and all. Issues 3 and 4 for this year will be combined into a large issue which I hope to be sending out before the end of the year. If you are working on an article for this newsletter, please try and get it to me before December 1 so that it can be edited in time.

Anthony Meadow, Editor

## About the Contributors

**Jay Rubin** is a professor of Japanese literature at the University of Washington. He has been processing words (specifically, the words of Natsume Sōseki's 1908 novel, Kōfu (The Miner)) since May 15, 1985 and can no longer believe that it was possible for him to translate Sōseki's Sanshirō (University of Washington Press, 1967) or to publish a study of prewar literary censorship, Injurious to Public Morals: Writers and the Meiji State (University of Washington Press, 1984) by writing in longhand and typing the manuscript on the portable typewriter he received as a high school graduation present. On the other hand, if he had spent all his time writing instead of making macrons on computers, he probably would have a couple more titles to his credit. He can be reached at 9821 N.E. 16th Street, Bellevue, WA 98004, or by phone at (206) 454-1473.

**George Hart** is a professor in the Department for South and Southeast Asian Studies at the University of California at Berkeley. He has several books to his credit. He can be reached at the Department for South and Southeast Asian Studies, 1203 Dwinelle Hall, University of California, Berkeley, CA 94720.

## Correspondence

Center for Southeast Asian Studies
Northern Illinois University
DeKalb, IL 60115

Dear Sir,

We have read with interest the article by David and Douglas Wyatt on the production of Southeast Asian scripts using their "universal typewriter." Your readers may be interested to know that we have thought the "immense" programming effort involved in more direct conversion of the romanisation to Southeast Asian othographies worth the effort required.

To date, we have developed precise, pronounceable romanizations and computer conversion programs for Tai Dam and standard Thai. The system is broadly generalizable to other Southeast Asian languages (Lao, Khmer, Burmese) and would probably require less than a month's effort for each additional language. Romanized, pronounceable text can be entered at the computer for immediate conversion and then be displayed on the computer's screen, printed on a dot matrix printer, or written as a binary image for later instantaneous screen display. Alternately, the text can be stored for later editing or conversion.

The software is written in Pascal, and runs on an Apple II computer with 64K or more. Printing may be done on an Apple DMP or Imagewriter. Modification of the software to drive other 8-pin dot matrix printers would not be difficult. An interactive character set editor has also be written to aid in the design of new character sets.

This system has been used to transcribe and print the Thai novel, "Red Bamboo" by Khukrit Pramot, and is also being used as a component of a computer-aided intruction project for Thai. Both native Thai speakers and American students have learned and used the romanization system.

Further details, program source code and documentation, and a disk containing the basic programs for Thai transcription are available from us through the Center for Southeast Asian Studies at Nothern Illinois University at nominal cost ($10 to cover production) for those interested in using and modifying the system on a non-commercial basis. A paper, "A Generalized Transliteration System for Southeast Asian Languages", describing the approach used in this system will appear in Seven Centuries Of Thai Writing to be published by The Center for Southeast Asian Studies in the fall of 1985.

Yours sincerely,

G. Henry (Computer Science, NIU) and
J. Hartmann (Foreign Languages and Literature, NIU)

## A Note on the Production of Macrons in Transliterated Japanese

by Jay Rubin
University of Washington

Tony K. Stewart did such a thorough job of setting out the problems involved in making diacritical marks in his "Diacritics on Wordstar: South Asian Language Transliteration Without Customized Software" (Volume 1, number 1 of this Newsletter), that there would be no point in reiterating them here. Since he specifically mentioned that he had not had the opportunity to test WordPrefect, the program I bought for its ability to deal with diacritics, and since, too, the methods I have developed seem to solve all the problems that remained with WordStar and would certainly be adaptable to South Asian languages, the results of my limited experience should be of interest to anyone still struggling with macrons or other such marks in word processing. They should be of the greatest use to those who have either not yet purchased hardware or, by coincidence, have bought hardware like mine. (I use an IBM PC compatible Compaq and a Brother daisy-wheel printer.) Even with these limitations, the technical data would be valuable to those working on other techniques for creating diacritics. In any case, as Tony Stewart noted with WordStar, it is essential to have a printer that will perform backspacing and superscripts, preferably one that can respond to codes adjusting the degree of its superscript/subscript movement (i.e. vertical movement) by 1/48th inch increments. I have also included the results of my efforts with Microsoft Word, a program comparable to WordPerfect. While they are less satisfactory, they may also prove useful.

### SSI's WordPerfect

WordPerfect (Version 4.0, 1985, by SSI Software, Orem, Utah) runs on the IBM PC and IBM PC compatible computers such as Compaq and Zenith. There are versions for other computers as well. The program contains both the backspace and superscript functions, allowing the user to place a raised hyphen over a character at any time while typing, although quite a number of keystrokes are required. WordPerfect also comes with a so-called macro facility which allows the user to save and recall virtually any series of keystrokes with the push of two keys (albeit with a three to four second delay while the computer recalls the macro from disk). By saving the keystrokes for "o", overstrike, superscript, and "-", and naming this combination "Alt-O", one can produce ō at any time simply by pressing the Alt key along with the O-key, which is no more complicated than making an upper-case letter on a typewriter. (One can just as easily save the keystrokes for "kūzen no daisōgi" and recall them with **Alt-K**, say.) Since WordPerfect is a "clean screen" word processor, the overstrike and superscript codes do not appear on the monitor unless (for editing purposes) the Reveal Codes key is pressed; normally, on the screen, "Sōseki" looks like "S-seki", the backspacing function causing the hyphen to cover the "o" until the "codes" are "revealed." Unfortunately, however, WordPerfect's superscript function moves a character up only one-third line, which brings the hyphen uncomfortably close to lower-case letters and makes it unusable for "i" and upper-case letters. The program also includes an "Advance Up" feature which moves characters up a full half line and an "Advance Down" to bring them down again, but this is not compatible with underlining, which "advances up" with the hyphen, cutting the letter in half like this: ō<u>gai</u>.

The solution to all of these problems (problems which must sound familiar to Tony Stewart's readers) lies in redefining the printer driver, i.e. the sub-program that translates the word processing program's generalized instructions into codes for the user's particular printer. SSI's excellent documentation and telephone support make this a straightforward procedure, one that requires no supplementary software. (The document PRINTER.MAN mentioned on page 15 of the Installation pamphlet and contained on the Supplementary diskette is a 33 page technical guide to

printer drivers and is well worth printing out.) Although the initial process is somewhat complicated, once you have created the character- with-macron combinations, they become a permanent part of the character font[1] in the word processing program and can be used at any time as easily as making a capital letter, without the delay involved in using a macro; they appear on the screen as quickly as any letter you type, and they are perfectly legible, having no distracting printer codes or symbols surrounding them, and no characters "hidden" by the backspace (or "overstrike") function. The steps for creating "ō" and "Ō", using the specific example of a Brother daisy wheel printer (models HR-15, -25, or -35) are as follows:

1. From the Set-up menu, as described in the Installation pamphlet, choose option 5, "Define Alt and Ctrl key mapping", and define **Ctrl-O** as decimal code 147 ( ô ), which is a pretty close equivalent to have on the screen. From now on, typing **Ctrl-O** will cause circumflex-o to be displayed, and 'Sōseki" will appear as "Sôseki". Exit WordPerfect.
2. Run the Printer Program, also following the instructions in the Installation pamphlet. Choose option 3, "Printer definition (Examine, Change)". From the "Printers Currently Defined" menu, choose option A, "Create". (Rather than risk changing anything in the given definition for your printer, you create a new definition modeled on it and play with that.) Enter any name you like for this new printer definition. Then designate which of the thirty-one current printer definitions you will use as a pattern: presumably, yours is among them. Select option 5, "Subscript/Superscript/Underline/Bold". Record the given codes for "Superscript on" and "Superscript off", but don't change them. Select item 0, returning you to the "Create" menu. Again select 0, returning you to "Printers Currently Defined". Exit.
3. From the main menu of the Printer Program, choose the character editing option (Number 4), and when you find ô in your particular character table (using the direction keys to scroll), replace the given string sent to the printer (probably "**o<8>^**", meaning the letter o followed by the character which has the decimal value of 8 (this is the standard backspace code) + a caret (shift 6 on most keyboards)) with a new string composed of o + backspace + a modified version of the "Superscript on" code + hyphen + "Superscript off" code. If the given "Superscript on" code **does** place the hyphen high enough to clear the letter, there is no need to modify it, but this can be done, as follows, on a Brother printer. That printer's code for "Superscript on" is **<27><30><3><27><10>**[2], but changing the **<3>** to **<4>** actually raises the hyphen just enough so that it leaves a nice space between the letter and the hyphen. In sum, then, the string sent to a Brother printer to produce ō is this: **o<8><27><30><4><27><10>-<10><B>**. Of course ū can be produced by replacing the given printer string for û with **u<8><27><30><4>**, etcetera. This will not work for "ī", as explained in step 4.
4. The **<4>** in the string sent to the printer is **not** high enough to clear upper-case letters or "i", but that problem is solved simply by using **<5>**. The string sent to the printer for Ō is **O<8><27><30><5><27><10>-<10><B>**, but because no standard code exists for upper-case O with a circumflex, we are left with the problem of finding an appropriate character for screen display. One choice in this case might be the Greek letter theta, but the oval face produced by decimal 002 code seems the right size and shape -- and is fun. In this case, **Alt-O** rather than **Ctrl-O** is "mapped" in step 1 to display the oval face on the screen since **Ctrl-O** was taken by ô.
5. Enter WordPerfect, hit the "Print" key; choose option 4, "Printer Control"; then option 3, "Select Printers"; and designate your newly defined printer as printer number 1.

---

[1] Actually, you are allowed to define up to six printers, each with eight fonts, so you can easily switch off these "permanent" characters whenever you like. If you want to write in French, you will not have sacrificed your circumflex-o by following the instructions below.

[2] See Appendix for other printers and codes.

Once we have completed the five steps, the O-key combined with Ctrl produces on the screen ô but prints ō, and combined with Alt it displays the oval face while it prints Ō. Since the macron is now "thought" by the printer to be part of the character, it never causes the printer to place underlining anywhere but where it belongs: under the character. And since the printer codes for backspace and superscript are now "built in" to the definition of the character, there is no delay while the computer retrieves a string of codes from disk: the process is instantaneous and not only far simpler than on any typewriter, it is far more refined, macron heights being custom-suited to each character. This adjustability of the degree of movement in superscripting was described by Tony Stewart as an exclusive feature of WordStar (and indeed, there is no mention of the capability in the WordPerfect documentation), but obviously that is not the case, and it can be especially useful to people in South Asian fields, who would want to modify both the "Superscript on" and "Subscript on" codes in this way to produce diacritics both above and below characters, a far more complicated task than that faced by us in Japanese fields.

## Microsoft's Word

Word (Version 2.0, 1985, by Microsoft Corporation, Bellevue, Washington) is another IBM word processing program, and it, too, comes in other versions. The manufacturer includes a 14 page document on the "Utilities Disk" called CONVPRD.DOC which, like WordPerfect's PRINTER.MAN, can be printed out and used as a technical guide to the program's printer drivers. Unfortunately, however, this document reveals only that Word is a far less adaptable program than WordPerfect when it comes to unusual tasks such as placing macrons over vowels. While displaying a circumflex-o on the screen is a relatively easy task (one need only look up the code, 147, in Appendix A, then type the code on the number pad while depressing the Alt key; no "mapping" of keys is necessary -- or possible), the only thing that will be produced by this on paper is circumflex-o. Word's printer drivers are not open and accessible like WordPerfect's and do not permit the user to substitute arbitrary strings of printer codes for the ones provided. Overstriking is the sole function of the drivers' "Character Translation Tables"; they cannot accomodate superscripting. Users of Word, then, are restricted to whatever can be accomplished from the keyboard. Since Word makes quite a lot accessible from the keyboard, this is not as restricting as it may sound, but the ease, speed, and typographical refinements are not available. The underlining of text containing macrons is also a serious problem, as outlined below.

One feature that makes Word simpler to work with is its definition of "Superscript" as a one-half (rather than one-third) reverse line feed: all our raised hyphens will be of the height necessary to clear both upper- and lower-case letters. On the other hand, once the superscript function is activated in Word, **all** subsequent characters are printed one-half line higher than before until the computer is told to return the printer to a "Normal" height. (Word's "Superscript" toggles on and off like WordStar's and is more like WordPerfect's "Advance Up" function; superscripting in WordPerfect works for only only character and turns off automatically.)

The steps necessary for producing any vowel with a macron, then, are as follows:

1. Type vowel.
2. Type **Alt-8**, using the 8 on the number pad, not the top row of typewriter keys. This produces a highlighted rectangle with a smudge in the middle of it, which is the symbol for the backspace code. (This symbol remains on the screen. "Sōseki" will appear as "So▮-seki").
3. Type a space, then press the left direction key to move the highlight one space to the left of the end mark. Otherwise, you'll be told in the next step that you can't edit the end mark.
4. Type esc-"Format-Character", Tab to "Normal-Superscript-Subscript", space bar to "Superscript". Press Enter.
5. Type hyphen.

6. Repeat step 4, but space bar to "Normal". Press Enter. (Note: if you are underlining, this step will turn off the underline function, and you will have to turn it on again. Better to add underlining afterward -- if at all[3].)
7. Press the right direction key, returning highlight to end mark. Backspace to remove extra space inserted in step 3.

Fortunately, Word allows the user to save keystroke combinations so that this procedure need not be repeated each time. The instructions for "Creating a Glossary" explain how the above steps can be stored and recalled, each combination of vowel-with-macron being given its own name. Assuming we give the name "o" to the combination "ō", we can insert "ō" into the text by typing esc-Insert-o-Enter. In order to have a permanent file of macrons in the Glossary. be sure to type "Y" for saving the Glossary before reconfirming that you want to Quit.

## Conclusion

Having devoted far too much of my life already to the question of how to place funny marks over letters, I should emphasize in conclusion that as little of one's writing time as possible should be taken up with getting the procedure to work. This, of all things, should be a "set it and forget it" matter. The system I have built around WordPerfect admirably fulfills this desideratum. If anyone can work out the underlining problem with Word, it should run a close second.

[3] Depending on the characteristics of your printer, the printer driver supplied by Microsoft may cause the underlining beneath text containing macrons to be shifted several spaces to the right. This can happen because Word does not distinguish between characters and the printer codes for overstrike and superscript; it "tells" the printer that the text to be underlined is somewhere other than where it really is. A Microsoft technician informs me that this can possibly be corrected by modifying bytes 144 and 150 of the printer driver, but I have had uneven results with this procedure.

## Appendix: Additional Printer Codes for WordPerfect

The following printers defined by WordPerfect use the same code for "Superscript on" as the Brother and would presumably respond to this modification in the same manner:

- C. Itoh Starwriter and Printmaster
- Diablo 630 ECS, 620 and 630 (Daisywriter)
- NEC 3515, 5515, 7715, 3525 and 7725
- Qume Sprint 5, 9, 11 and 11+
- Texas Instruments 855
- Toshiba P1350 and P1351
- Dataproducts DP series

The following are other printers with their very different codes for "Superscript on". Note that a zero looks like 0 and capital o looks like O:

| | |
|---|---|
| • Centronics 351: | **<27>L** |
| • Epson FX, MX-Graftrax and MX Type III: | **<27>S<0>** |
| • HP 2686A LaserJet-B: | **<27>&a-35V<O><4>** |
| • IBM Wheelprinter and 5218: | **<27>A<2><27>2<27>]** |
| • IBM Quietwriter: | **<27>S<0>** |
| • NEC 2050 and 3550: | **<27>A<3><27>2<27>7** |
| • NEC 3510, 3530, 7710 and 7730: | **<27>]R<27>9** |
| • Okidata Microline 93: | **<27>J** |
| • Toshiba P1340: | **<27>VP@B** |

Some of these look more promising than others; the only printer I can be sure of is the Brother.

## Indian Fonts on the Macintosh

by George L. Hart
University of California, Berkeley

As my field is classical South Asian languages, and as I work with Sanskrit and Tamil every day, I have long been interested in ways to get the scripts for those languages (and for other South Asian languages) on paper in a legible form. This problem has been exacerbated by the fact that I must prepare vocabulary for students, and my handwriting is terrible -- students cannot read my English handwriting, much less my Tamil or Devanagari. Thus, when microcomputers started appearing in large numbers about five years ago, I bought an Apple II and began to learn how to use it.

After a great deal of work, I was able to overcome one problem -- I wrote a program that produced the proper diacritical marks for South Asian languages -- but I still was unable to put Tamil or Devanagari onto a printed page.

Then, about a year and a half ago, the Macintosh appeared. It was immediately apparent that that machine was different in radical ways from its predecessors, and I set out to discover whether it would be able to produce Devanagari and Tamil more easily than other machines. I was fortunate enough to get in contact with a senior programmer at Apple, Mark Cutter, who wrote LisaDraw and MacDraw (which are shape-based drawing programs), and who himself was interested in implementing Devanagari. In fact, I discovered that Mark had already developed a Devanagari that runs on the Lisa, the predecessor of the Macintosh. While his implementation had some rough edges -- which was inevitable, considering he did not know any Indian languages -- it was still remarkably good, and it showed that Devanagari could be made to work on the Lisa and on its successor, the Macintosh. What made the system especially appealing was that Devanagari did not appear only on the printed page; it appeared on the screen also. In fact, the screen and the printed page were identical.

Since that time, Mark and I have been engaged in a project to develop and perfect South Asian scripts on the Macintosh. The rest of this article concerns what we have discovered, the progress we have made to date, and what we hope to do in the future.

It is extremely difficult to produce Indian scripts in a satisfactory manner on most computers. The reason for this is that the shapes of the characters are stored in ROM and cannot be changed. Of course, most computers -- the Apple II, the IBM PC, and most others -- either have or can be given a graphics mode, and it is possible to write programs that produce whatever character one wishes on the screen in graphics mode when a key is pressed. The screen can subsequently be dumped to a printer, giving a print-out of the characters one has written in.

There is one disastrous shortcoming to this procedure. A good word-processing program (or other sort of program in which one might wish to use non-Roman characters) takes many man-years to produce. The graphics routines that are developed in the above procedures cannot be accessed through any standard word-processing programs. Rather, whoever uses them must write his own word-processor which, then, must be rudimentary, as he does not have the resources of a large software producer at his disposal -- he cannot have several people working full-time on a processor for, say, Hindi. Thus, one is left with a rudimentary process that lets one put Devanagari or another alphabet on the screen and to print it out, but which lacks the capabilities that make computers really useful: the ability to manipulate text in a complex fashion, to use advanced data base techniques, etc.

The Macintosh differs from these computers in several ways, but in one respect that is crucial for those who wish to use non-Roman alphabets. Everything the computer does on the screen is graphics oriented. On most microcomputers, when you press "a" on the keyboard, the machine looks in ROM somewhere, retrieves the pattern corresponding to "a" and puts it at the current screen position. On the Macintosh, when you press "a", the operating system looks to see what the current font (character set) is, what the current character size is (usually 9, 12, 18, or 24 points), and what the current character attribute is (normal, bold, italic, shadowed, outlined,

underlined, or a combination of these). If it does not have that font already in memory (RAM), it loads it in from a file called "system", and then proceeds to print the character in the desired size with the desired attributes on the screen, using routines built into the operating system to scale the character, underline it, italicize it, or whatever.

Most Macintosh programs have the ability to select the font you wish; in the major word-processing programs, it is possible to change fonts, sizes, etc. at any point. Thus it is an easy thing to have a page with different styles of type, different sizes of type, and even different alphabets. All of the most complex functions (search, replace, justification, footnoting, fancy formatting) of word processing are available in all alphabets

The process of implementing Devanagari, Tamil, or any other Indian alphabet that is left-to-right is a relatively simple one. It consists of defining a matrix mapping (bit-map) for each character, deciding the width and placement of the character, deciding a unique number and name for the font, and then putting that information, all coded in the right format called a "resource", into the operating system (which is contained in a file called "system"). Fortunately, there exist programs called font editors that accomplish these tasks in a relatively painless manner. One uses the mouse to fill in squares on the screen and so create the character, and decides on the width and placement of the character by manipulating little pointers with the mouse.

Still, Indian alphabets have certain special problems, and it was important to see whether the Macintosh could accomodate them. First, most alphabets (except for Tamil) have a large number of characters -- ligatures and other variants -- that need to be made room for. The keys on a typewriter are not adequate for this, and as a result Devanagari typewriters do not allow you to type all the characters found in a printed book. Fortunately, the Macintosh gives access to twice as many keys as a normal keyboard: one can type all of the regular keys (lower and upper case), and then by pressing the "option" key, type another entire set of characters in lower and upper case. On the normal Macintosh Roman keyboard, not all of these "option" keys are defined, but they can be defined by any font editor and used for such scripts as Devanagari. For those readers who know the basics of computers, it is worth remarking that this extra set of keys is implemented by paying attention to the high bit on the byte representing the character. This gives double the number of characters that can be represented by the simple ASCII system (which pays attention to only 7 bits and ignores the 8th).

Another problem is that of characters that must go over other characters. In Devanagari, for example, "pa" is p while "pe" is ep (in Tamil, cf. g and gP). If it is impossible to overwrite either preceding or succeeding characters, it will not be possible to write an "e" that comes after a consonant in Devanagari or a long "i" that comes there in Tamil. Fortunately (and somewhat unexpectedly), the system that Apple chose to translate fonts to the screen (a system that it took partly from the Xerox research group in Palo Alto) handles this peculiarity of Indian scripts perfectly. It is possible to designate both the width and position of a character within very broad limits, so that all of the combinations encountered in Devanagari and Tamil can be handled easily.

Once it was apparent that it was practical to implement Indian fonts on the Macintosh, the next step was to actually do it. First of all, a keyboard layout had to be decided upon. This, of course, was not as critical as it would have been in designing a typewriter, as on a computer it is possible to reassign keys relatively easily. For Tamil (which possesses no ligatures or voiced sounds), this was an easy task: that language has many fewer characters than Devanagari and has a fairly standard implementation on a typewriter; all we had to do was to copy the typewriter keyboard, with a few changes that made typing on the computer simpler. For Devanagari, the situation is very different. First, it has almost twice as many characters as Tamil, and second it seems to have an endless number of keyboards designed for it over the years (we have two different layouts on the two Hindi typewriters in our department, and I have diagrams of another six or seven in my files). Several years ago, I had typed my Sanskrit textbook on an IBM Devanagari typewriter, and had grown partial to that layout, which puts the Devanagari sounds in the same position as the sounds on a Roman keyboard (asdfg equals asdfg). The advantage to this keyboard is that it is easily learned by someone who types English -- a case that includes most people who would want to type Hindi, Marathi, or Sanskrit on the Macintosh.

The next step was to actually implement the fonts, a process that was hampered by lack of a good font editor for the Macintosh (which made it necessary to do font development on the Lisa; there is now an excellent font editor for the Macintosh marketed by Altsys Corp. of Plano, Texas). I began with Tamil, as its alphabet is much simpler than Devanagari, and then took the Devanagari already done by Mark Cutter and modified it to conform to the IBM keyboard and to the needs of Sanskrit and Hindi. Since then, development of these fonts has been continuing, both by me and by others. The following things can be reported:

1. Apple has developed a new "keycaps" desk accessory, which allows one to have a picture of the keyboard for any font one is using on the screen in a window next to the document he is typing. This makes it easy to learn to type in an Indian alphabet, as the keyboard layout can be consulted as often as necessary.
2. M.G. Srinivasan developed an improved Devanagari taking the work of Mark Cutter and myself as a model. He has called this Kanchi; I am now engaged in adding punctuation to this alphabet and modifying it to make it easier to type. When this is finished, Hindi should be very nearly as easy to type as English.
3. Using a product called Thunderscan -- which digitizes images from paper or photographs into the format of Macintosh (Macpaint) drawings -- I was able to read in Tamil characters and use them to create a much more esthetic Tamil alphabet than my original. I hope to have this completed for distribution (along with the new Devanagari) in two or three months.
4. Professor Donald Becker of the University of Wisconsin, who has done extensive work in creating software that prints out Devanagari, Telugu, and Arabic from the IBM PC onto a Toshiba printer (the screen being in Roman, the printing in the desired alphabet), has developed a Gurmukhi font for the Macintosh. He has just recently sent me a preliminary version of Telugu for the Mac, along with a manual that describes how to use it.
5. Zhang Liansheng, a visiting lecturer at Berkeley from Peking, has developed a Tibetan script for the Macintosh.
6. Two different Bengali scripts have been developed. One, by Dr. M. Z. Iqbal of Caltech, is being distributed on the public-domain font disk.
7. Professor Grant Olson of Cornell University has developed a font for Thai.

There are several things that remain to be done. First, we need to get more Indian alphabets on the Macintosh. Equally important, we must make it possible to print the fonts on the Laserwriter, Apple's new laser printer that produces output of publishable quality. The difficulty with this is that the LaserWriter does not store fonts as bitmaps (as the Macintosh itself does), but as shapes (like Macdraw). It has software that allows it to create a bitmap of any size from a given shape, thus insuring the best possible resolution for each character (the LaserWriter has 300 dots per inch). When we have Indian fonts on the Laserwriter, it will be possible to compose a book on the Macintosh using Hindi, Tamil, English and other Indian alphabets in different sizes, styles, and mixtures, use the Imagewriter (Apple's dot-matrix printer) to get it into final shape, and then to use the Laserwriter to print out the final version.

The Macintosh's operating system is designed to be amazingly flexible. It is possible, for example, to designate any font one wishes as the system font. This means that the entire computer (or any specific program) can be made to come up and interact with the world in Hindi, Tamil, or Marathi instead of English. Programs on the Macintosh store the messages and menus they use in special resources that can be modified using a resource editor. This means that one can take a sophisticated program like the Word (which does word processing) and modify it easily to function entirely in, say, Tamil. All of the messages and text the program uses would then appear in that language. In other words, with not too much work, the Macintosh and almost all of the programs developed for it could be converted to any Indian language that we have developed a font for. The potential of this for such things as elementary school education is enormous.

It should be emphasized that even now, the same Macintosh and the same program can be used to print a letter in Devanagari, Tamil, English, Gurmukhi, Bengali, Telugu, or Tibetan, or in any

combination of those languages; one does not even have to change disks. Ultimately, all of the left-to-right Indian scripts should be available for the Macintosh. Indians from different parts of the country who work in the same office could produce materials in all the major languages except Urdu: different language versions of an official document or form could be prepared and printed on the same machine; a Bengali and a Tamil could write letters in their respective languages without even changing the program they are running. One hopes that this universality of the Macintosh with regard to different languages and scripts will contribute to the integration of the India.

To receive copies of the Devanagari and Tamil fonts for the Macintosh, send a check for $10 made payable to the Regents of the University of California to:

Professor George Hart
Department of South and Southeast Asian Studies
1203 Dwinelle
University of California
Berkeley, CA 94720

## Examples of South Asian Fonts on the LaserWriter

A Sanskrit verse from the *Gita:*

आपूर्यमाणमचलप्रतिष्टं समुद्रमापः प्रविशन्ति यद्वत् ।
तद्वत्कामा यं प्रविशन्ति सर्वे स शान्तिमाप्नोति न कामकामी ॥

*A person whom all desires enter*
*as the waters enter the ocean which, being filled,*
*remains without motion --*
*he attains peace, not the man*
*who runs after desire.*

Two lines from the Tamil *Purananuru:*

யாதும் ஊரே யாவரும் கேளிர்
தீதும் நன்றும் பிறர் தர வாரா

*All lands home, all men kin;*
*evil and good do not come from others.*

Some lines in Telugu:

విజయనగరచక్రవర్తి కృష్ణదేవరాయల కాలం ఆంధ్ర
సంస్కృతికీ ఆంధ్ర సాహిత్యానికీ సువర్ణ యుగం.

*The time of Krishnadevaraya, the Vijayanagara king, was the golden age of Andhra culture and Andhra literature.*

A verse in Punjabi:

ਸਾਬਣ ਲਾ ਲਾ ਧੋਤਾ ਕੋਲਾ, ਦੁਧ ਦਹੀਂ ਵਿਚ ਪਾਇਆ ।
ਖੰਬ ਚਾੜ ਰੰਗਣ ਭੀ ਧਰਿਆ, ਰੰਗ ਨਾ ਉਸ ਵਟਾਇਆ ।
ਵਿਛੜ ਕੇ ਕਾਲਖ ਸੀ ਆਈ, ਬਿਨ ਮਿਲਿਆਂ ਨਹੀਂ ਲਹਿੰਦੀ ।
ਅੰਗ ਅੰਗ ਦੇ ਲਾ ਕੇ ਦੇਖੋ, ਚੜ੍ਹਦਾ ਰੂਪ ਸਵਾਇਆ ।

(ਭਾਈ ਵੀਰ ਸਿੰਘ)

*I tried to wash a piece of coal with soap and put it into milk and curd to make it white, but to no effect. Then I processed it with different dyes to color it but it did not change its color. It had become black on separation from its beloved, that would not go unless it is reunited. Put it in fire and you will find it glowing with all its beauty.*
*(Bhai Veer Singh)*

## Reviews of Books, Journals and Articles

Note that addresses are included for some publications which come from smaller or lesser known publishers. The editor has written all the reviews in this section.

### Books

Geoffrey James, **Document Databases**, Van Nostrand Reinhold, New York, 1985. (ISBN 0-442-28185-4). This book presents a simple, non-technical overview of document databases. No attention is given to languages other than English and there are only a handful of references to papers and other books. This book is of little or no use to the readers of this newsletter.

H. S. Hou, **Digital Document Processing**, Wiley & Sons, New York. This book gives a technical overview of the topic described by the title. The chapters are:

1. Introduction
2. Input Scan
3. Shape Manipulation
4. Digital Halftoning and Shading
5. Data Compression
6. Character Feature Extraction
7. Document Preparation
8. Document Retrieval
9. Text Recognition
10. Document Transmission
11. Output Scan

This is a rather technical book written by a staff scientist at Xerox Corporation. Each chapter includes a several page bibliography which makes this a good source book for those needing an overview. It was obviously intended to give only an overview of the topics covered, but it is nevertheless a worthwhile book for those interested in topics such as OCR (Optical Character Recognition) and so on.

Joan Knoerdel, **A Survey of Standardization Efforts of Coded Character Sets for Text Processing**, National Bureau of Standards, Report Number NBS SP 500-81, 1981. (Library of Congress Number: 81-600108). Available from the US Government Printing Office, Washington, DC 20402). This technical report describes the standard character sets used in the United States. The basic character set is known as ASCII, and this report gives the details of it and other related character sets. It also describes the standards organizations that generate them, as well as new standards under way for "Page Description Languages". These are general codes that are used to describe items such as page size, margins, headers and footers, etc. that in the future should allow the direct exchange of documents from any word processing system to another.

Roy Andrew Miller, **The Japanese Language**, University of Chicago Press, Chicago, 1967, reprinted as a Midway Reprint in 1980. (ISBN 0-226-52718-2) This book is an introduction to the history and structure of Japanese. Chapter 3, *Writing Systems*, will be of interest to readers of this newsletter who need a reference on this topic.

G. Sampath, **An Introduction to Text Processing**, River Valley Publishing, P.O. Box 99752, Jeffersontown, KY, 40299, 1985. (ISBN 0-9615070-0-4) The subtitle reads: a systematic approach to the study of text structure and operations and the design of text processing software. It is based on a graduate level course in Text Processing taught by the author at Syracuse University and the University of Louisville. The contents are (by section and chapter):

I. Overview
    1. The Text Processing System
II. Texture
    2. Character Processing

3. String Processing
4. Text Storage and Retrieval - Files and Buffers
5. Text Patterns - Tokens, Words, Paragraphs and Segments
6. Text Patterns - Creation and Modification: Text Entry and Editing

III. Form

7. Display Form - Extraction and Generation: Formatting and Formatted Editing
8. Syntactic Form - Extraction and Generation: Lexical Analysis, Parsing and Syntactic Editing

IV. Content

9. Content Processing: Statistical Analysis, Context, Semantics and Usage

V. Interfaces

10. Text Compression
11. The User Interface

Bibliographic notes are at the end of each chapter and a rather large bibliography is included as Appendix C. This is the only technical book that I have ever seen that deals with the basic issues of text processing (excluding more sophisticated topics such as computational linguistics). This is an excellent book and is highly recommended for anyone interested in designing and implementing text processing systems.

T.C. Ting (editor), **Chinese-Character Processing for Computerized Bibliographic Information Exchange**, International Development Research Centre, Box 8500, Ottawa, Canada K1G 3H9. (ISBN 0-88936-441-9, IDRC-239e). This is a summary report of an international workshop held in Hong Kong on December 17-20, 1984. There are five short articles and a 15 item bibliography. It is primarily of interest to librarians, and developers of systems for librarians.

The following technical reports on TEX and METAFONT (see the previous issue for more on these two programs) are available from the Stanford Computer Science Department. They can be ordered from: Publications, Computer Science Department, Stanford University, Stanford, CA 94305; telephone: (415) 497-4776. California residents should add 6.5% sales tax.

| Report | Author | Title | Cost |
|---|---|---|---|
| CS-824 | Tung | "LCCD - a language for Chinese character design" | M |
| CS-828 | Knuth, Plass | "Breaking paragraphs into lines" | M |
| CS-848 | Tang | "On the problem of inputting Chinese characters" | M |
| CS-870 | Plass | "Optimal pagination techniques for automatic typesetting systems" | M |
| CS-886 | Knuth | "Concept of a Meta-font" | 2.35 |
| CS-901 | Fuchs, Knuth | "Optimal font caching" | 2.60 |
| CS-914 | Gu, Hobby | "Using string matching to compress Chinese characters" | 2.50 |
| CS-960 | Zabala | "Interacting with graphic objects" | 4.95 |
| CS-965 | Ghosh | "An approach to type design and text composition in Indian scripts" | 6.45 |
| CS-966 | Bigelow, Ghosh | "A formal approach to lettershape description for type design" | 3.60 |
| CS-974 | Gu, Hobby | "A Chinese meta-font" | 2.70 |
| CS-977 | Liang | "Word Hy-phen-a-tion by com-put-er" | 4.70 |
| CS-978 | Knuth | "Lessons learned from METAFONT" | 3.05 |
| CS-980 | Knuth | "The WEB system of structured documentation" | 8.10 |
| CS-981 | Knuth | "Literate Programming" | 2.45 |
| CS-985 | Samuel | "First Grade TEX" | 3.95 |
| CS-1013 | Désarménien | "How to Run TEX in French" | 3.30 |
| CS-1027 | Knuth | "A Torture Test for TEX" | 9.25 |

If the cost is described above as "M", then the report is available in microfiche. All other reports are available on both paper and microfiche. Each report in microfiche costs $2.00. Many of these reports do not deal with languages other than English, but are listed here because they may still be of some interest. I have not read any of these reports yet.

**The Proceedings of the 1985 International Conference on Chinese Computing**, published by the Chinese Language Computer Society, is now available for $40 (US and Canada) or $45 in the rest of the world. It can be ordered from: C.N. Liu, CLCS Treasurer, 18-114, IBM Research Center, PO Box 218, Yorktown Heights, NY 10598. I have not seen it yet.

## Journals

**Byte** magazine, $21/year (US), $23/year (Canada, Mexico) with higher rates for other countries is available from: Byte Subscriptions, PO Box 597, Martinsville, NJ 08836. William Raike writes "Byte Japan", a regular column in **Byte** magazine, one of the more well known microcomputer magazines. He describes many of the new Japanese computers, some (but not all) of which will be of interest to those working with Japanese on computer. There are other articles that appear in it that are also of interest.

The journal **Computer Processing of Chinese and Oriental Languages** is published by the Chinese Language Computer Society, which was described in the previous issue of this newsletter. Copies of the journal are available from: Wellington C.P. Yu, CLCS Membership Chairman, General Products Division, A63/098, IBM Corporation, 5600 Cottle Road, San Jose, CA 95193. The contents of the first five issues are:

**Volume 1, issue 1 (July 1983):**

J. Hobby and G. Gu, "Using Metafont to design Chinese characters", pp. 4-23.
S. Mori, "Research on machine recognition of handprinted characters", pp. 24-39.
Y. Chu and L.S.-C. Chan, "Chinese micro-COBOL: a language for Chinese data processing", pp. 40-58.
T.Y. Huang, C.F. Wang and Y.H. Pao, "A Chinese text-to-speech synthesis system based on an initial-final model", pp 59-70.
G.N. Ni and P. Tanner, "The application of anti-aliasing technique for displaying high quality Chinese characters", pp. 71-77.
X. Wang, "System design of a computerized and laser typesetting system for Chinese characters", pp. 78-85.

**Volume 1, issue 2 (December 1983):**

S.C.Lee, S. Xu and B. Guo, "Microcomputer-generated Chinese speech", pp. 87-103.
T.C. Yu, "Voice input to computers", pp. 104-115.
P.L. King, "Human factors and linguistics: keys to high speed Chinese data entry", pp. 116-123.
J.C. Tsay and I.C. Wu, "Computer-assisted page composition of a Chinese newspaper", pp. 124-134.
J.M. Unger, "A mnemonic code for Sino-Japanese characters (Kanji) based entirely on their readings", pp. 135-144.
K. Chai, "A combination of statistical and fuzzy set theory applied to recognition of handprinted Chinese characters", pp. 145-149.

**Volume 1, issue 3 (May 1984):**

S. Matsubayashi and H. Hagiwara, "Mass production of Kanji fonts and inhouse printing/publishing system", pp. 153-162.
C.Y. Suen and E.M. Huang, "Computational analysis of the structural compositions of frequently used Chinese characters", pp. 163-176.
Y. Liu, "Language engineering in China", pp. 177-185.
J.W. Tai, "A syntactic-semantic approach for describing Chinese characters", pp. 186-194.
Y.H. Lee and C.H. Kwok, "A database implementation of an interactive Chinese telephone directory", pp. 195-204.
S.K. Chang, D. Zhang and J. Stigler, "A computerized abacus for office automation and computer-aided instruction", pp. 205-210.

**Volume 1, issue 4 (November 1984):**

S.K. Wan, H. Saitou and K.I. Mori, "Experiment on Pinyin-Hanzi conversion Chinese word processor", pp. 213-224.
W.C.P. Yu and T.C. Chen, "Two-level encoding for Chinese input systems", pp. 225-235.
C.K. Chen and R.W. Gong, "Evaluation of Chinese input methods", pp. 236-247.
H.R. Hwa and C.Y. Chung, "A new Chinese coding method for information processing", pp. 248-265.
H.W. Hwang and T.Y. Hwang, "Microcomputer CAI for Chinese language with character generator", pp. 268-275.
I.M. Liu, "Recognition of fragment-deleted characters and words", pp. 276-287.

**Volume 2, issue 1 (May 1985):**

C.H. Chen and Y.H. Pao, "Computer recognition of Chinese language (Mandarin) homonyms", pp. 1-22.

W.C. Lin and T.T. Luo, "Synthesis of Mandarin by means of Chinese phonemes and phoneme-pairs (JIFH)", pp. 23-35.
S.Y. Lo, "A scientific model for comparing various methods of inputting Chinese characters into computer", pp. 36-58.
J. Yang, "A psychological view on the standardization of the structural elements of Chinese characters in information encoding", pp. 59-70.

Occasionally there are interesting articles in **Infoworld**, a popular computer news weekly. It was formerly published in a magazine format but has recently returned to its initial tabloid form. Several articles that are of interest are:

Alexander Besher, "Hong Kong's microcomputer industry", 12 March 1984, pp. 79-81.
David Rothman, "Saudi Arabia Embraces Micros", 23 April 1984, pp. 25-26.
Peggy Watt, "Legends Inscribed on Floppies", 21 May 1984, pp. 25-26.
Alexander Besher, "Japan on 16K a Day", 28 May 1984, pp. 66-68.
Judith Walthers, "Translators Gain Fluency", 13 August 1984, pp. 35-37.
Mark Fruin, "T/Maker makes inroads into the Japanese market", Vol 5(49), pg. 75.

**Language Monthly** is a periodical that seems aimed primarily at translators and language teachers. They feature articles on translating, machine translation, trends in schools and universities as well as reviews of dictionaries and periodicals. It is available for £17 per year in England and $28 per year in the US. Orders should be sent to:Praetorious Limited, 5 East Circus Street, Nottingham NG1 5AH, England.

**Research in Word Processing** is a new newsletter. It is published to be a "clearinghouse of information pertaining to computer-based writing instruction at all educational levels - from elementary to Ph.D." There will be nine issues per year and it will feature original research, article abstracts, bibliographies and reviews. It is available for $12 per year in the US from: The Editors, Research in Word Processing Newsletter, South Dakota School of Mines and Technology, Rapid City, SD 57701-3995. I have seen only one issue so far. It looks very good, especially for people interested in using computers in education.

**Technical Japanese Translation**, a newsletter mentioned in the previous issue of this newsletter, has ceased publication.

**Visible Language** is a quarterly that deals with the visual aspects of language expression. It is available for $15 per year (for individuals) from: Visible Language, Box 1972, c/o The Cleveland Museum of Art, Cleveland, OH 44106. There has been a special issue edited by J. Marshall Unger entitled: "Aspects of the Japanese Writing System", **Visible Language** 18(3), Summer 1984. Articles in this issue are:

Chris Seeley, "Aspects of the Japanese Writing System: Introduction", pp. 213-218.
A.E. Backhouse, "Aspects of the Graphological Structure of Japanese", pp. 219-228.
Nannette Twine, "The Adoption of Punctuation in Japanese Script", pp. 229-237.
J. Marshall Unger, "Japanese Orthography in the Computer Age", pp. 238-253.
J. Marshall Unger, "Japanese Braille", pp. 254-266.
Chris Seeley, "The Japanese Script since 1900", pp. 267-301.

## Articles

Elhanan Adler, "Judaica Automation in Israel - An Overview", **Judaical Librarianship** 1(1), pp. 9-11 (Fall 1983). This article describes all the current "Judaica on computer" projects in Israel. Brief descriptions of the various systems are given.

Joseph Becker, "Typing Chinese, Japanese and Korean", **Computer**, January 1985, pp. 27-34. This article discusses the nature of the difficulties in typing in the ideographic languages and how it

has been implemented on the Xerox workstations. **Computer** is published by the Institute of Electrical and Electronics Engineers in the US.

Kathleen Burton, "Over There: IBM In Japan", **PC Magazine**, July 1983, pp. 276-280. A non-technical article on IBM in Japan, especially on the IBM 5550 Japanese PC.

Douglas Cooper, "The Computerization of East Asian Languages", **Library Software Review** 3 (2), pp. 179-186. This article reviews the Research Library Group's and OCLC / Asiagraphics's CJK (Chinese - Japanese - Korean) systems. There is also a small bibliography with several articles from trade journals on CJK systems and related topics.

Jere Fleck, "Multi-language Use: A Survey of the Problem", **Capital PC Monitor** 4(7), pp. 38-42. This article discusses the problems of multi-lingual users on IBM PCs. It was published in the Capital PC Users Group newsletter. I don't have an address for them, but they are the largest IBM PC Users Group in the Washington, D.C. area.

David Gilner, Ellen Kovacic and Hebert Zafren, "Hebrew Cataloging via Microcomputer", **Small Computers in Libraries** 3(9), pp. 3-5 (Sept 1983). This article describes a system developed for the library at the Hebrew Union College to catalog books in Hebrew. A sample card is shown. The software runs on an Apple II.

David Gilner, Ellen Kovacic and Hebert Zafren, "Hebrew Cataloging at Hebrew Union College on an Apple II+", **Judaical Librarianship** 1(1), pp. 4-7 (Fall 1983). This article describes the system developed for the library at the Hebrew Union College mentioned in the above article. There is a little more technical detail than in the previous article, as well as some discussion of its use in everyday (library) tasks.

Timothy Huang, "First Chinese Forth - A Double-Headed Approach", **Dr. Dobb's Journal**, issue 94 (June 1984). This article describes an implementation of the Forth computer language in Chinese. See two letters about this article in **Dr. Dobb's Journal**, issue 95 (September 1984), pp. 8-10. **Dr. Dobb's Journal** is one of the best technical microcomputer magazines published. Subscriptions are available for $25 per year (in the US) from: Dr. Dobb's Journal, PO Box 27809, San Diego, CA 92128.

Daniel H.H. Ingalls and Daniel H.H. Ingalls, Jr., "The *Mahabharata*: Stylistic Study, Computer Analysis, and Concordance", **Journal of South Asian Literature** 20(1), pp. 17-46. This article de - scribes an ambitious project to study the text of the *Mahabharata* by means of computer. It describes a successful method for scanning and recognizing Devanagari (OCR).

Masasuke Morita, "Japanese Text Input System", **Computer** 18(5), pp. 29-35 (May 1985). This article describes a new approach to typing Japanese into a computer. The new approach requires a new keyboard that is organized along the same principles as the Dvorak keyboard. **Computer** is published by the IEEE Computer Society, a branch of the primary association for electronic engineers in the US.

James Nye, "Indic Fonts for Computer Printers", **South Asian Library Notes and Queries**, issue 18 (Spring, 1985). This brief article lists several sources of Indic fonts, primarily for the Apple Macintosh, but some for other machines as well. **SALNAQ** is published by the Committee on South Asian Libraries and Documentation of the Association of Asian Studies's South Asia Council. An annual subscription costs $8 and is available from: South Asia Library Notes and Queries, c/o South Asia Collection - Room 560, University of Chicago Library, 1100 East 57th Street, Chicago, IL 60637.

Recently, it seems as though every computer magazine has reviewed most or all of the keyboard "macro" programs such as ProKey. These programs are of interest to all readers who use IBM PCs or compatibles, as Tony Stewart's article in the previous issue of this newsletter demonstrated. Three such articles that I've run across recently are:

David Obregón, "Power Plays At Your Keyboard", **PC Magazine** 4(22), pp. 167-175 (Oct 29, 1985). Reviews ProKey, SuperKey, Keyworks, RE/Call, Newkey and Smartkey.

John Walkenbach, "Keyboard Shortcuts", **PC Tech Journal** 3(10), pp. 131-144 (Oct 1985). Reviews ProKey, SuperKey, Keyworks, RE/Call, Smartkey and Keyswap.

Dan Swearingen, "In the Right Key", **PC Products** 2(10), pp. 33-49 (October 1985). Reviews ProKey, SuperKey, Keyworks, RE/Call, SmartKey and Keyswap.

Now that Apple's LaserWriter and other laser printers are out, there is more interest in typography. These printers are dramatically changing our views of what good quality output looks like. Here are four articles on this topic. Note that even though some of them are now several years old, they are still quite relevant.

Charles Bigelow, "Font Design for Personal Workstations", **Byte** 10(1), pp. 255-270 (Jan 1985). Discusses the difficulties of designing fonts for computer workstations.

Charles Bigelow, "Digital Typography", **Scientific American** (Aug 1983). Discusses the basics of digital typography and how fonts have evolved from older methods of setting type.

Jonathan Seybold, "Digitized Type: What is it? What does it mean for typesetting and word processing?", **The Seybold Report** 8(24), pp. 3-17 (27 Aug 1979). Gives an excellent presentation of what digital typography is all about.

Charles Bigelow and Jonathan Seybold, "Aesthetics vs. Technology: Does Digital Typesetting Mean Degraded Type Design?", **The Seybold Report** 10(24), pp. 3-16 (24 Aug 1981). Gives an explanation of what a typographic designer looks for when designing a typeface, as well as a brief history of typesetting.

**The Seybold Report** was described in the previous issue of this newsletter.

The following articles are all the articles found in all issues to date of the journal **Computer Science and Informatics** that deal with Indian scripts on computer. This journal is published by the Computer Society of India, described in the previous issue of this newsletter. It was formerly known as the **Journal of the Computer Society of India**. This is a difficult journal to locate in the United States. For those living in the San Francisco Bay area, the only library that has a complete set of back issues is the Mathematics Library on the Stanford University campus. The articles are:

S. Andres and S. Ramani, "A Note on Programming a Character Generator for the Devanagari Script", JCSI 1(1), pp. 55-56. Software for Calcomp plotters.

K.P.S. Menon, "Direct Input/Graphical Output of Two Indian Languages - Hindi and Malayam", JCSI 2(1), pp. 14-26. Description of FORTRAN programs that produce output on Calcomp plotters. Has samples of output of both languages.

P. Narasimham, B. Prasada and V. Rajaraman, "Code Based Keyboard for Indian Languages", JCSI 2(2), pp. 33-37. Gives description of current and proposed keyboards for Indian languages.

S.N.S. Rajasekaran and B.L. Deekshatulu, "Generation and Recognition of Telugu Characters", JCSI 4(2), pp. 3-6. Describes proposals to generate and recognise Telugu characters.

D. Dutta Majumder, A.K. Dutta and S.K. Pal, "Computer Recognition of Telugu Vowel Sounds", JCSI 7(1), pp. 14-20 (1976). Description of a system developed to recognise Telugu vowel sounds. The system was developed on a Honeywell 400 computer.

S.N.S. Rajasekaran, "Computer Generation and Recognition of Printed Telugu Characters", JCSI 7(1), pp. 42-43 (1976). Short description of the author's Ph.D. thesis.

S.N.S. Rajasekaran and B.L. Deekshatulu, "Generation of Artificial Character Database", JCSI 8(2), pp. 49-54 (1978). Description of method of generating test data for OCR work.

S.K. Agarwal and H.N. Mahabala, "Character-ROM Based Display for Indian Languages", JCSI 11(1), pp. 9-14 (1981). Describes a hardware system for generating Indian scripts.

R. Subbu, "Review of 'Phototypesetting in Indian Languages: Design Information Reports for Four Scripts'", CS&I 12(1), pp. 12 and 27 (1982). Briefly reviews these reports which would apparently be useful to anyone designing scripts for Indian languages.

S.P. Mudur and R. Sujata, "Three Systems for Typesetting: A Survey", CS&I 12(1), pp. 28-36 (1982). Reviews DIP (from NCSDCT, Bombay, India), SCRIBE and TEX - three typesetting software systems.

A special issue of the **Journal of the Institution of Electronics and Telecommunication Engineers** devoted to "Computer Applications in Processing of Indian Languages and Scripts" has come out. This special issue was edited by Dr. R.M.K. Sinha. Copies of this issue may be ordered from: The Editor, The Institution of Electronics and Telecommunication Engineers, 2 Institutional Area, Lodi Road, New Delhi 110 003, India. The titles of all articles and letters are as follows:

**Articles:**

R.M.K. Sinha, "Computer Processing of Indian Languages and Scripts - Potentialities and Problems", pp. 133-149.
R. Chandrasekaran, M. Chandrasekaran and Gift Siromoney, "Computer Recognition of Tamil, Malayam and Devanagari Characters", pp. 150-154.
A.K. Datta, "A Generalized Formal Approach for Description and Analysis of Major Indian Scripts", pp. 155-161.
J.B. Millar, R.K. Barz and A.V. Diller, "Document Production in Hindi and Thai via a Software Man-Machine Interface", pp. 162-167.
R. Gupta, A. Banerjee and S.K. Mullick, "Coding of Devanagari Composite Character Patterns for Data Compression", pp. 168-172.
P.L. Emiliani, P. Graziani and A. Tronconi, "Speech Synthesis for Italian Language and its Applications", pp. 173-178.
Behrooz Parhami, "Standard Farsi Information Interchange Code Keyboard Layout: A Unified Proposal", pp. 179-183.
N.R. Ganguli and A.K. Datta, "On Machine Recognition of Telugu Speech Sounds", pp. 184-189.
T.K. Ghoshal, Gourhari Das, K.K. Datta, S. Mitra and S. Bhattacharya, "Vidyasagar - A Bengali-Ahamia Text Processing Attachment", pp. 190-195.
P.K. Ghosh, "Basic Design Issues in a Multi-Lingual Type Font Design and Typesetting Workstation", pp. 196-202.
C.V.K. Singh, S.V. Rangarajan and P. Seetharamaiah, "Alphanumeric CRT Terminal for Telugu and Hindi Scripts", pp. 203-214.
A. Nageswara Rao, "New Indo-Roman Script for Transliteration of all the Indian Languages", pp. 215-222.

**Letters to the Editor:**

T.R. Rammohan and B.N. Chatterji, "Recognition of Distorted Kannada Characters", pp. 223-225.
A.K. Ray and B. Chatterjee, "Design of a Nearest Neighbour Classifier System for Bengali Character Recognition", pp. 226-229.
S. Mitra, S. Bhattacharya and T.K. Ghoshal, "Representing Variable Width Composite Consonant Text in Character Mode Raster-Scan VDU", pp. 229-230.
A.N. Sinha and G. Kashipati Rao, "Release Burst and Formant Transition as Acoustic Cues for English Stop Consonants Perception by Native Speakers of Indian Languages", pp. 231-233.
S. Raman, R. Sundar and H.N. Mahabala, "Microprocessor-Based Stencil Cutter as a Computer Output for Text in Indian Languages", pp. 234-237.
Donald Becker, "Romanizing Hindi as a Prerequisite for Word Processing", pp. 238-242.
R.M.K. Sinha and B. Srinivasan, "Machine Transliteration from Roman to Devanagari and Devanagari to Roman", pp. 243-245.
R.M.K. Sinha, "Segment Display for Devanagari Script and Numerals", pp. 246-248.
R.M.K. Sinha and K.S. Singh, "A Program for the Correction of Single Spelling Errors in Hindi Words", pp. 249-251.
Gaurhari Das, S. Bhattacharya and S. Mitra, "Representing Ahamia, Bengali and Manipuri Text in Line Printer and Daisy-Wheel Printer", pp. 251-256.
Zhang Liansheng, "A Design for the Computerized Processing of Tibetan Scripts", pp. 257-260.

This is an excellent collection of articles on Indian languages on computer. While there is some chaff among the wheat, this issue is highly recommended to anyone working with Indian languages on computer.

## Product Listings

In this section, we will report on hardware and software products that are of some interest to the readers of this newsletter. Note that none of these items is being reviewed. All information has been supplied by the vendor. If someone has used any products here and writes a review, we would be delighted to print it. If you are a vendor of hardware or software that would be of interest to the readers of this newsletter, you are also invited to submit an article for publication.

## Hardware Products

### Xerox Corporation: 6085 and 8010 Star Workstations

Description: Computer systems with specialized software allowing word processing in twenty languages, including Chinese, Japanese and Korean.

These workstations are the ones that started the revolution in user interfaces popularized by the Apple Lisa and Macintosh computers, and since copied by the new user interface programs for the IBM PC such as IBM's TopView, Digital Research's Global Environment Manager (GEM) and Microsoft's Windows.

The development of the user interface continues unabatedly at Xerox and the Star workstation family now supports word processing and other functions in 20 languages, including English, French, German, Russian, Chinese, Japanese and Korean. Instead of having twenty different keyboards, the system requires only one, with a "virtual keyboard" displayed on the screen in the current language. Any number of languages can be used in a document. Some technical details of how this difficult task was accomplished have been detailed in several papers by Joseph Becker (see the Reviews section of this and the previous issue of this newsletter).

The workstations support intermixed text and graphics, tables and forms. Each workstation has a large (1024 pixels horizontally by 808 pixels vertically), high resolution display (72 pixels per inch horizontally and vertically). The keyboard has a standard typewriter layout and 24 additional function keys. There is also, of course, a mouse. There are several configurations of the machine with varying amounts of hard disk storage and networking capabilities. A set of workstations can be connected together by an Ethernet LAN (local area network).

Japanese can be entered using either roman characters (romanji) or kana. If characters are entered in roman characters, they are automatically converted to kana. When a word is completed, the system tries to locate all Kanji characters that have the same spoken sound as that just typed in. The user is then presented with the choices in order of highest frequency of use.

Chinese is entered by typing in the sounds of the word in either pinyin or bopomofo. Tone numbers can also be entered to speed the typing process. The system then looks up all possible Chinese words that have that same sound and the choices are again presented to the user.

For more information, contact Xerox Office Systems Division, 2100 Genge Road, Palo Alto, CA 94303; telephone: (415)496-6170.

### CPT: Phoenix 8525 Word Processor

Description: Stand-alone word processing system for English and Arabic.

CPT is a major stand-alone word processing vendor. They have an Arabic package for their CPT 8525 word processor. This package allows the intermixing of English and Arabic in the same

document. The software automatically selects the appropriate form of each character as it is typed. Documents can be printed on either their standard dot matrix printer or daisy-wheel printer.

For more information, contact CPT Corporation, 8100 Mitchell Road, PO Box 295, Minneapolis, MN 55440. Telephone: (612) 937-8000.

### Eastern Computers Inc.: Graphics Boards

Description: Apple II and IBM PC compatible boards for word processing and working with Chinese (other languages to follow).

Eastern Computers, Inc. has developed series of boards that allow an Apple II or IBM PC (or compatible) to display and work in various languages. The first board can display more than 30,000 Chinese characters in a 16 by 16 dot matrix. The boards will allow programming in Chinese as well as word processing. Additional boards for other languages, as well as improvements (including a 24 by 24 dot matrix), are under development.

For additional information, contact: Eastern Computers, Inc., 600 Lynnhaven Parkway, Virginia Beach, VA 23452; telephone (804) 340-2496.

### Fujitsu: My OASYS Word Processor

Description: Stand-alone word processing system for Japanese and English.

Fujitsu is now selling their My OASYS Japanese word processor in the United States. The machine displays Japanese characters in a 24 by 24 pixel format. A basic set of 3418 kanji is stored in the machine (the basic JIS first level set of kanji) and it can store an additional 3384 user defined kanji. Text can be entered in either kana or romanji. Conversion to kanji is performed through user interaction. The system can also work in English. Basic graphs can also be generated using the machine. A dot matrix printer is provided for output.

For more information, contact: Catherine Kawakami, Fujitsu America, Inc., 3055 Orchard Drive, San Jose, CA 95134; telephone: (408) 946-8777.

### Maracom: The Font Machine

Description: Hardware/software system based on an IBM AT microcomputer for developing fonts for printers.

Maracom Corporation has developed a font development system based on the IBM PC/AT. This system can be used to develop fonts for a wide range of output devices including dot matrix printers, laser printers and other types of "dot" matrix printers. Fonts can be scanned in using a video camera. The characters are then edited on a video monitor. Dots can be moved, added, removed and portions of characters can be moved.

The system was developed primarily for printer manufacturers and is available only on a contract basis, i.e. not "off-the-shelf."

For additional information, contact: Maracom Corporation, 648 Beacon Street, Boston, MA 02215; telephone (617) 266-3630.

### Multitech: DCS570 Workstation

Description: Microcomputer system for word processing and programming in English and Chinese. Uses an 8088 CPU (same as IBM PC) and a version of Digital Research's Concurrent DOS.

Multitech has developed a bilingual workstation that supports both the Chinese and the English languages. A hardware character generator holds 17,000 Chinese characters and displays them in a 24 by 24 dot matrix. The operating system is a Chinese version of Digital Research's Concurrent DOS, which allows multi-tasking and windows.

Additional software available includes a Chinese version of the T/Maker database package. The system is based on an 8088 CPU and includes 512Kbytes of RAM, two floppy drives, a serial and a parallel port, a keyboard and a monitor and has room for 6 IBM PC compatible slots for additional cards. The base price is $6250. Several options are also available.

For additional information, contact: Michael Tsai at Multitech Electronics Inc., 195 W. El Camino Real, Sunnyvale, CA 94086, telephone (408) 773-8400 in the US; or William Lu at Multitech Industrial Corp., 266 Sung Chiang Road, 9F, Taipei 104, Taiwan, R.O.C., phone (02)551-1101.

### Research Computer Technology Corp.: Alraed Computer Series

Description: CP/M and MS-DOS compatible microcomputers for word processing and programming in Arabic and English. Applications software for both languages is available.

Research Computer Technology Corporation has developed a series of three CP/M compatible computers that allow programming and use in Arabic and English. All three models are based on the Z-80 CPU chip. A more recent model, the Alraed 600, has an Intel 8086 and an Intel 80286 CPU. A letter quality daisy-wheel printer is available. A wide variety of software is available, including many programming languages, such as Basic, C and COBOL. Applications software is available for both the Arabic and the English languages.

For more information, contact: Research Computer Technology Corporation, 3152 Kashiwa Street, Torrance, CA 90505; telephone (213) 534-5800.

## Software Products and Services

### BISA: BISA Database

Description: An on-line database service available world-wide that covers Southeast Asian materials in Australian libraries.

The BISA (Bibliographic Information on Southeast Asia) project was started in the 1970's to provide more access to the collection of Southeast Asian material in Australian libraries. A comprehensive electronic database is available for use from virtually anywhere in the world. Indonesia is the country with the most coverage in the database, with Malaysia and Singapore running second. The database can be searched by author, title, subject or any individual keyword, and the results can be restricted by language, publisher, date of publication, etc.. The database stores information in any language.

BISA also publishes books and monographs on material from the catalog. They also run a traineeship program for librarians on database creation and information retrieval.

For more information, contact: Helen Jarvis, Director BISA, University of Sydney, NSW 2006, Australia; telephone: 692-2222 or 692-3164 (direct line).

### Apanda: Chinese Word Processing System

Description: Apple II based word processing system for Chinese.

Apanda has developed a Chinese word processing system based on the Apple II computer. An IBM PC version of the software is under development as well as Japanese Hiragana and Katakana and Korean characters set. Characters are displayed in a 24 by 24 matrix, but a larger size of 48 by 48 is also supported, making higher quality output possible by photoreduction. There are 6000 simplified Chinese characters provided along with 7700 classical characters.

Character input is via a unique code assigned to each character. Additional character entry methods will be available in the future.

For more information, contact: Apanda, Inc.,5650 Kirby Drive, Suite 249, Houston, TX 77005; telephone: (713) 661-2114.

### Altsys Corporation: Fontastic Font Editor

Description: Font editor for Macintosh computers.

Altsys Corporation has developed a font editor for the Apple Macintosh computer. This editor allows a user to modify an existing font or create a new font. The fonts can then be installed by the user. These fonts are then usable as any other Macintosh font, i.e. they can be used in any word processor or applications program that supports the use of fonts. The font editor allows copying of fonts to and from MacPaint files. In the latest release (version 2.0), a bit-level paste and cut operation is available. This means that portions of a character can be easily moved to other characters, a great convenience when many characters have the same basic forms. The program allows editing of fonts up to 120 points in size.

The Fontastic font editor replaces the Apple font editor which is available through CompuServe and other channels. Fontastic has several advantages over the Apple editor: it doesn't crash (or appear to have any bugs) and there are more facilities for editing fonts.

Altsys has announced a new product which will be available in December this year - a font designer that will create LaserWriter fonts. This is a very exciting product that will allow the creation of fonts, logos and so on that can potentially look professional. Macintosh fonts are simply bit maps of varying height and length. LaserWriter fonts are described by a set of curves, and therefore look good no matter what size they are enlarged or reduced to (assuming of course that the font designer does a good job).

For more information, contact: Altsys Corporation, PO Box 865410, Plano, TX 75086; telephone: (214) 596-4970.

### Asiagraphics: The Asiagraphics System

Description: Word processing software for Chinese and Japanese for HP-86 microcomputers .

Asiagraphics has developed a multi-lingual system based on the Hewlett-Packard HP-86 computer. The system allows use in Chinese, Japanese and English. They can be intermixed on the same line. The system can store up to 20,000 separate characters.

Chinese can be entered through the pinyin, bofomofo or Wade-Giles system. The user enters the pronunciation of each word and the system searches for the possible characters that have the same sound. The user then selects the correct character. Four levels of operation are available,

allowing more experienced users to work at a faster rate. New characters can be created by means of an interactive character design program.

This system can be used with the OCLC on-line library database system.

For additional information, contact: Asiagraphics, 141 Mt. Sinai Avenue, Mt. Sinai, NY 11766; telephone: (516)246-8365.

### CEEDE: Instructional Materials

Description: Instructional software packages for the Apple II microcomputer in Vietnamese, Hmong and Lao.

The Center for Educational Experimentation, Development and Evaluation at the University of Iowa offers instructional materials in Vietnamese, Lao and Hmong (as well as many other programs and projects). All programs run on Apple II+ or IIe computers. The programs are for language instruction and cultural study programs.

Many other unrelated instructional software packages and training programs are also available.

For more information, contact: CEEDE, N 345 Oakdale Hall, Oakdale, IA 52319; telephone: (319) 353-4200.

### Data Transforms: Fontrix

Description: Apple II and IBM PC software package to develop and print fonts.

Data Transforms offers a graphics software package for the Apple II and the IBM PC that allows a much larger variety of fonts and graphics forms than usual. The program can be used to design forms, tables and even maps. Any text file can be read in by the program, which may pose some interesting problems in integrating another word processor with this program. It works with a wide variety of dot-matrix printers. A font editor is included as part of the package.

They also sell sets of fonts. Among the fonts available are: Arabic, Hebrew, Japanese (Hiragana and Katagana) and Sanskrit (including all the vowels but only consonants in the form consonant + short a).

A review of this product appeared in the 21 January 1985 issue of Infoworld, on pages 41-43.

For more information, contact: Data Transforms, 616 Washington Street, Denver, CO 80203; telephone: (303)832-1501.

### Gulf Data: Arabrite and Arabdos

Description: Word processing and programming packages for IBM PC compatible computers.

Gulf Data offers two programs: the Arabrite word processing program and Arabdos programming tool. Both operate on the IBM PC and compatibles.

The Arabrite word processing program allows the entry of both Arabic and English, even on the same line. The program automatically determines the correct form of each Arabic character. The commands, menus, help screens and manual are in both Arabic and English. It can print using either dot-matrix printers or single or dual track letter quality printers. Stickers for the keyboard are provided.

The Arabdos programming tool allows programmers to develop applications in Arabic, English or a combination of the two languages on the MS-DOS operating system. It will work with the Basic, COBOL, C, Fortran and dBase II programming languages. Examples of programs in several of these languages are provided. Applications developed can: read in English or Arabic text, perform the insertion of Arabic text into English and vice versa, display Arabic and/or English

prompts, update Arabic or English data files and output Arabic or English files to dot matrix or letter quality printers. Arabdos also supports downloadable character sets, allowing variations in characters.

If a display is needed for displaying Arabic, an IBM PC compatible board is available.

For more information, contact: Gulf Data, Inc., 9015 Fulbright Avenue, Chatsworth, CA 91311; telephone: (213) 998-0922.

### ICL: Hanyupinyin Word Processor

Description: Word processing software package for Chinese on IBM PCs and compatibles. .

International Computerized Linguistics has developed a Chinese language word processor based on the IBM PC. The system will run on an IBM PC or any compatible with a color graphics adapter. Many different dot matrix printers are supported. A character editor is provided so that additional characters may be drawn by the user. Characters are developed in a 24 pixel by 16 pixel matrix. Text can be entered in pinyin or Mandarin along with the vowel tones.

For more information, contact: International Computerized Linguistics, Inc., 5885 South Gessner, Houston, TX 77036; telephone: (713) 266-2179.

### Lexisoft: Spellbinder/Arabic

Description: Word processing software that runs on IBM PC compatible microcomputers for word processing in Arabic and English.

Lexisoft has developed a word processing system that works with both Arabic and English. The program runs on IBM PCs and most compatibles. The languages can be intermixed in a document. The program automatically decides the correct form of each Arabic character.

They also offer a similar product, Spellbinder Scientific, that allows scientists to enter mathematical and chemical formulae in documents. It comes with a graphics editor that can be used to design additional characters.

For more information, contact: Lexisoft, Inc., PO Box 1378, Davis, CA 95617; telephone: (916) 758-3630.

### Proper Software: The Font Librarian

Description: Font moving program for the Apple Macintosh.

Proper Software is distributing a font moving program for the Macintosh. This program replaces the Apple Font/Desk Accessory Mover that is available through CompuServe and other channels. The Font Librarian allows a user to look at a font without having to install the font. Fonts can be examined in several ways: as a string of characters, as an ASCII chart, as a chart marked with the key caps that produce each key, or the user can type in any text desired. The text can be scaled to any size, and viewed with any combination of style options (bold, underline, shadow, etc.). Fonts can be renamed, renumbered, copied and deleted. Any number of fonts can be viewed at the same time, whether they are in the same file or not.

The Font Librarian is a "shareware" product, that is, anyone can copy it or give it to friends, but if you use the program, you should send in a registration fee. In exchange, you receive a disk and a manual for the program.

For more information, contact: Proper Software, 2000 Center Street, Suite 1024, Berkeley, CA 94704; telephone: (415) 540-5958.